Administrator's Guide

iPlanet Web Server, Enterprise Edition

Version 4.1

806-4641-01
March 2000


Recycled and Recyclable Paper

Documentation Team: Jocelyn Becker, Robert Fish, Ann Hillesland, Sanborn Hodgkins, Amanda Lee, Laila Millar, Alan Morgenegg, and June Smith

Version 4.1

Printed in the United States of America. 00 99 98 5 4 3 2 1

Contents

Part 4 Using Programs and Objects

Appendixes

About This Guide

This guide describes how to configure and administer iPlanet™ Web Server. It is intended for information technology administrators in the corporate enterprise who want to extend client-server applications to a broader audience through the World Wide Web.

This preface includes the following sections:

- What's In This Guide?
- How This Guide Is Organized
- Conventions Used In This Guide
- Using the iPlanet Web Server Documentation
- Further Reading
- Contacting Technical Support

What's In This Guide?

This guide explains how to configure and administer the iPlanet Web Server. After configuring your server, use this guide to help maintain your server.

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

How This Guide Is Organized

This guide is divided into five parts, plus various appendices, a glossary, and a comprehensive index. If you are new to iPlanet Web Server, begin with Part I, "Server Basics" for an overview of the iPlanet Web Server. If you are already familiar with iPlanet Web Server, skim the material in Part I, "Server Basics" before going on to Part II, "Using the Administration Server."

Once you are familiar with the fundamentals of using the Administration Server, you can refer to Part III, "Configuring and Monitoring," which includes examples of how to configure and monitor your iPlanet Web Servers. Part IV, "Using Programs and Objects" provides information for using programs and configuration styles. Part V, "Managing Content and Access" provides information for managing your iPlanet Web Server content, controlling access to your iPlanet Web Servers, how to use Web Publisher to collaborate on projects, and how to search the contents and attributes of documents on your servers.

Finally, the appendices address specific reference topics that describe the various topics, including: HyperText Transfer Protocol (HTTP), server configuration files, ACL files, internationalization issues, server extensions, and the iPlanet Web Server user interface reference, which you may want to review. Note that the user interface appendix is available in the online version only.

Part I: Server Basics

This part provides an overview of the iPlanet Web Server. The following chapters are included:

- Chapter 1, "Introduction to iPlanet Web Server," provides an overview of iPlanet Web Server.
- Chapter 2, "Administering iPlanet Web Servers," describes how to manage your iPlanet Web Servers with the Administration Server.

Part II: Using the Administration Server

This part provides conceptual and procedural details using the Administration Server to administer your iPlanet Web Servers. The following chapters are included:

- Chapter 3, "Setting Administration Preferences," describes how to use the Administration Server Preferences and Global Settings forms to configure your iPlanet Web Servers.
- Chapter 4, "Managing Users and Groups," describes how to how to use the Administration Server Users and Groups forms to configure your iPlanet Web Servers.

- Chapter 5, "Working with Server Security," describes how to configure your iPlanet Web Server security. Note that before reading this chapter you should be familiar with the basic concepts of public-key cryptography and the SSL protocol. These concepts include encryption and decryption; keys; digital certificates and signatures; and SSL encryption, ciphers, and the major steps of the SSL handshake. For more information regarding these topics, see *Managing Servers with Netscape Console*.

- Chapter 6, "Managing Server Clusters," describes the concept of clustering servers and explains how you can use them to share configurations among servers.

Part III: Configuring and Monitoring

This part includes examples of how to use the Server Manager to configure and monitor your iPlanet Web Servers. The following chapters are included:

- Chapter 7, "Configuring Server Preferences," describes how to configure server preferences for your iPlanet Web Server.

- Chapter 8, "Understanding Log Files," describes how to monitor your iPlanet Web Server using the Hypertext Transfer Protocol (HTTP), by recording and viewing log files, or by using the performance monitoring tools provided with your operating system.

- Chapter 9, "Using SNMP to Monitor Servers," describes how to monitor your iPlanet Web Server using SNMP (Simple Network Management Protocol).

- Chapter 10, "Configuring the Server for Performance," describes how to define your server workload and sizing your system to meet your performance needs. This chapter addresses miscellaneous configuration and Unix/Linux platform-specific issues, CGI-related performance tuning problems, and other common performance issues.

Part IV: Using Programs and Objects

This part provides information for using the Server Manager to programs and configuration styles. The following chapters are included:

- Chapter 11, “Extending Your Server With Programs,” describes how to install Java applets, CGI programs, JavaScript applications, and other plug-ins onto your server.
- Chapter 12, “Working With Configuration Styles,” describes how to use configuration styles with iPlanet Web Server.

Part V: Managing Content and Access

This part provides information for using the Server Manager to manage your iPlanet Web Server content, control access to your iPlanet Web Servers, how to use Web Publisher to collaborate on projects, and how to search the contents and attributes of documents on your servers. The following chapters are included:

- Chapter 13, “Managing Server Content,” describes how you can configure and manage your server’s content.
- Chapter 14, “Controlling Access to Your Server,” describes the methods you can use to determine who has access to what files or directories on your web site.
- Chapter 15, “Configuring Web Publishing,” describes how you can configure iPlanet Web Server for web publishing.
- Chapter 16, “Using Search,” describes how to search the contents and attributes of documents on the server. In addition, this chapter describes how to create a customized text search interface that’s tailored to your user community.

Appendixes

This section includes various appendixes for reference material that you may wish to review. This section includes the following appendixes:

- Appendix A, "HyperText Transfer Protocol," provides a short introduction to a few HTTP basic concepts.
- Appendix B, "ACL File Syntax," describes the access-control list (ACL) files and their syntax.
- Appendix C, "Internationalized iPlanet Web Server," describes the internationalized version of the iPlanet Web Server.
- Appendix D, "Server Extensions for Microsoft FrontPage," describes using server extensions on your iPlanet Web Server that provide support for Microsoft FrontPage.
- Appendix E, "iPlanet Web Server User Interface," describes the elements in the user interface of the Administration Server and Server Manager of iPlanet Web Server. This appendix is available in the online version only.

In addition, a glossary is included to define frequently used terms that may be unfamiliar to iPlanet Web Server administrators.

Conventions Used In This Guide

The conventions used in this guide are as follows:

Italic — 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.

bold — Bold style is used for new terminology and specific dialog box and drop down menu options. All new bold terms are also in the glossary.

Using the iPlanet Web Server Documentation

The following table lists the tasks and concepts that are described in the iPlanet Web Server printed manuals and online readme file. If you are trying to accomplish a specific task or learn more about a specific concept, refer to the appropriate manual.

Note that the printed manuals are also available as online files in PDF and HTML format.

Table 0.1 iPlanet Web Server Documentation

For information about	See the following
Late-breaking information about the software and the documentation.	http://www.iplanet.com/docs
Installing iPlanet Web Server and migrating your data to the new iPlanet Web Server.	*iPlanet Web Server Installation & Migration Guide*
Administering one or more iPlanet Web Servers using the Administrator Server to manage and configure your servers and to perform the following tasks: • Setting up server security. • Monitoring your servers using HTTP, via log files, SNMP, or via the tools provided with your OS. • Defining your server workload and sizing your system to meet your performance needs. • Installing Java applets, CGI programs, JavaScript applications, and other plug-ins onto your server. • Configuring iPlanet Web Server for web publishing. • Searching the contents and attributes of server documents; creating a text search interface.	*Administrator's Guide*
The administration server and global information on topics such as encryption, access control, and performance monitoring.	*Managing Servers with Netscape Console*

Table 0.1 iPlanet Web Server Documentation

For information about	See the following
Planning your directory service. How you can use the directory server to support simple usage that involves only a few hundred users and some key server applications, as well as how you can scale the directory server to support millions of users. You are also introduced to the basic directory service concepts and specific guidelines that you will need to deploy a production-grade directory service.	*Netscape Directory Server Deployment Manual*
Using the web publishing system. This manual is included with your server in HTML format.	*Netshare and Web Publisher User's Guide*
An overview of the programming technologies and APIs you can use to extend and modify iPlanet Web Server, to dynamically generate content in response to client requests, and to modify the content of the server. Links are provided to the individual books that discuss each API. This book also contains information about API changes from Enterprise 3.x to iPlanet Web Server 4.x. Use this book as the starting place for developer-level information for iPlanet Web Server 4.x.	*Programmer's Guide to iPlanet Web Server*
How to enable and implement servlets and JavaServer Pages (JSP) in iPlanet Web Server.	*Programmer's Guide to Servlets in iPlanet Web Server*
How to use Netscape Server Application Programmer's Interface (NSAPI) to build plugins to extend and modify the iPlanet Web Server. The book also discusses the purpose and use of the configuration files `obj.conf`, `magnus.conf`, and `mime.types`, and provides a comprehensive list of the directives and functions that can be used in these configuration files. It also provides a reference of the NSAPI functions you can use to define new plugins.	*NSAPI Programmer's Guide for iPlanet Web Server*
How to create Server-Side JavaScript (SSJS) applications. JavaScript is Netscape's cross-platform, object-based scripting language for client and server applications.	*Server-Side JavaScript Guide*

Further Reading

The iPlanet Documentation site contains documentation for administrators, users, and developers, including:

- iPlanet Web Server *Release Notes*
- *JavaScript Reference*
- Netscape Internet Service Broker programmer's guides and reference guides for Java and C++
- *Web Publishing Client API Guide*

To access these documents, use the following URL:

```
http://www.iplanet.com/docs
```

Contacting Technical Support

For Technical Support assistance, please see the Technical Support Page for the iPlanet Web Server at:

```
http://www.iplanet.com/support/
```

1

Server Basics

- **Introduction to iPlanet Web Server**
- **Administering iPlanet Web Servers**

Chapter

1

Introduction to iPlanet Web Server

This chapter introduces iPlanet Web Server and discusses some of the fundamental server concepts. Read it to obtain an overview of how iPlanet Web Server works.

This chapter includes the following sections:

- iPlanet Web Server
- iPlanet Web Server Architecture
- How iPlanet Web Server is Configured
- Administration Server
- Server Manager
- Netscape Console
- Sending Error Information

iPlanet Web Server

iPlanet Web Server is an extremely powerful multi-process, multi-threaded, secure web server built on open standards that enables your business enterprise to seamlessly integrate with other internal and external systems. By

providing high performance, reliability, scalability, and manageability, iPlanet Web Server solves the business-critical needs of your web site, regardless of the size of your enterprise.

This section includes the following topics:

- iPlanet Web Server Features
- Administering and Managing iPlanet Web Servers

iPlanet Web Server Features

iPlanet Web Server is primarily designed to provide access to your business HTML files. In addition, it offers the following features:

- **Web publishing**—End users can organize and publish their documents from their desktops with a web publishing interface. They can organize documents by type to customize presentation for different purposes, and use text search to manage document content via the Content Management (CM) feature. CM is an NSAPI plug-in that allows you to manage files on a remote server, with drag and drop like capabilities (via a web publishing applet) and index document content in an intelligent way for easier content searching.

- **Enterprise-wide manageability**—Including delegated administration, cluster management, and LDAP (Lightweight Directory Access Protocol) support. LDAP integration with Netscape Directory Server enables you to store users and groups in a centralized directory. In addition, 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.

  Note that in order to add users and groups to iPlanet Web Server, you must have a directory server installed, such as Netscape Directory Server. If you need to create, locate, or manage records for users and groups on any other servers within your network, you should use Netscape Console with your Directory Server. For more information, see *Managing Servers with Netscape Console*.

- **Security**—Users can establish encrypted and authenticated transactions between clients and the server through the Secure Sockets Layer (SSL) 3.0 protocol. In addition, iPlanet Web Server employs the following security-

based standards: Public Key Cryptography Standard (PKCS) #11, which defines the interface used for communication between SSL and PKCS #11 modules; Federal Information Processing Standards (FIPS)-140; and special certificates that work with 40, 56, or 128 bits, depending on the capability of the client.

- **Access control**—You can protect confidential files or directories by implementing access control (viewing, editing, and version control) by username, password, domain name, or IP address. This feature also represents another aspect of the NSAPI Content Management plug-in, which enables an end user (the owner of a document) to set access control on a document, rather than having to ask the administrator to accomplish the task.

- **High performance**—Delivers high performance for dynamic and secure content with features such as HTTP1.1, multi-threading, and support for SSL hardware accelerators.

- **Standards-based**—iPlanet Web Server includes support for a wide range of web software standards, including: JDK 1.2; Servlets 2.1; JavaServer Pages 1.1; HTTP 1.1; and various security-based standards, including PKCS #11, FORTEZZA, FIPS-140, and 128-bit step-up certificates.

- **Server-side Java Servlet and JavaServer Pages support**—enables development of server plugins, dynamic content, presentation logic, and JDBC database access.

- **Server-side JavaScript support**—enables development of scripting applications that access the database using native drivers.

- **Additional features**—Support for multiple processes and process monitors, failover, automatic recovery, and dynamic log rotation.

Administering and Managing iPlanet Web Servers

You can manage your iPlanet Web Server(s) via the following user interfaces:

- iPlanet Web Server Administration Server

- Server Manager

- Netscape Console

In previous releases, the Enterprise Server and other Netscape servers were administered by a single server, called the Administration Server. In the 4.x release, the "administration server" is now just an additional instance of the iPlanet Web Server, called **iPlanet Web Server Administration Server**, or **Administration Server**. You use the Administration Server to administer all of your iPlanet Web Server instances. For more information, see "Administration Server."

Note You can also perform administrative tasks manually by editing the configuration files or by using command-line utilities.

For managing individual instances of iPlanet Web Server, you can use the Server Manager. For more information, see "Server Manager."

If you have other 4.x iPlanet Web Servers, you can manage them through the Netscape Console, a client-based Java application. For more information, see "Netscape Console" or *Managing Servers with Netscape Console*.

iPlanet Web Server Architecture

iPlanet Web Server incorporates a modular architecture that integrates seamlessly with all of the products in the Netscape/iPlanet family of servers. You can use the Netscape Console when you need to perform administrative functions across all of the Netscape/iPlanet servers. In addition, the iPlanet Web Server includes an administration server interface for coordinating administrative functions across all of your web servers. Note that this administrative interface is itself another instance of iPlanet Web Server.

iPlanet Web Server includes the following software modules:

- Content Engines
- Server Extensions
- Runtime Environments
- Application Services

These server modules are described in the following sections.

Content Engines

iPlanet Web Server content engines are designed for manipulating customer data. The following three content engines make up the Web Publishing layer of the iPlanet Web Server architecture: HTTP (Web Server), Content Management, and the Search (Verity).

The **HTTP engine** represents the core of the iPlanet Web Server. From a functional perspective, the rest of the iPlanet Web Server architecture resides on top of this engine for performance and integration functionality.

The **Content Management engine** enables you to manage your server's content. You create and store HTML pages, JavaServer Pages, and other files such as graphics, text, sound, or video on your server. When clients connect to your server, they can view your files provided they have access to them.

The **Search engine** enables iPlanet Web Server users to search the contents and attributes of documents on the server. As the server administrator, you can create a customized text search interface that works with various types of documents formats, such as HTML, Microsoft Word, Adobe PDF, and WordPerfect. iPlanet Web 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.

Server Extensions

The iPlanet Web Server extensions enable you to extend or replace the function of the server to better suit your business operations. The following server extensions are part of the core iPlanet Web Server architecture:

- Common Gateway Interface (CGI)
- Netscape Server Application Programming Interface (NSAPI)
- Java Servlets and JavaServer Pages
- SHTML & JavaScript

Common Gateway Interface (CGI) is a stand-alone application development interface that enables you to create programs that process your client requests dynamically.

Netscape Server Application Programming Interface (NSAPI) is used to implement the functions the server calls when processing a request (Server Application Functions) which provide the core and extended functionality of the iPlanet Web Server. It allows the server's processing of requests to be divided into small steps which may be arranged in a variety of ways for speed and flexible configuration.

Java Servlets and JavaServer Pages extensions enable all Java servlet and JavaServer page meta-functions, including instantiation, initialization, destruction, access from other components, and configuration management. Java servlets and JavaServer pages, are reusable Java applications that run on a web server rather than in a web browser.

SHTML and Server-side JavaScript enable rapid development of dynamic content applications.

Runtime Environments

In addition to the various server extensions, iPlanet Web Server includes a set of runtime environments which support the server extensions. These runtime environments include the following:

- CGI Processor
- NSAPI Engine
- Java Virtual Machine (JVM)
- JavaScript Virtual Machine

Application Services

Finally, the iPlanet Web Server architecture includes a set of application services for various application-specific functions. These application services include the following:

- LiveWire Database Service
- Security & Access Control
- Session Management Service
- File System Service
- Mail Service

How iPlanet Web Server is Configured

iPlanet Web Server is configured to enable you to turn on or off various features, determine how to respond to individual client requests, and write programs that run on and interact with the server's operation. The instructions (called directives) which identify these options are stored in **configuration files**. iPlanet Web Server reads the configuration files on startup and during client requests to map your choices with the desired server activity. For more information about these files, see "iPlanet Web Server Configuration Files."

The server includes a number configuration files which are stored in *server_root*/config when installed on your computer.

This section includes the following topics:

- iPlanet Web Server Component Options
- iPlanet Web Server Configuration Files
- Single-Server Configuration
- Multiple-Server Configuration

iPlanet Web Server Component Options

The following component options are available when you install iPlanet Web Server:

- iPlanet Web Server Core
- Java Runtime Environment
- Java and Servlets
- ServerSide JavaScript Database Connectors
- Web Publishing
- SNMP

iPlanet Web Server Configuration Files

iPlanet Web Server includes a variety of configuration files that enable you to set various global variables, and to customize how the server responds to specific events and client requests. You can modify the configuration files automatically using the Administrator Server or Server Manager user interface settings, or manually by editing the files directly. For more information, see Chapter 10, "Configuring the Server for Performance."

The main iPlanet Web Server configuration files are: `magnus.conf`, `obj.conf`, `mime.types`, and `admpw`. These configuration files are described in this section.

Note There are a number of configuration files iPlanet Web Server uses when your server is set up as part of a cluster of iPlanet Web Servers (these files include a `.clfilter` file extension). For more information regarding how you can configure a cluster of iPlanet Web Servers, including important guidelines, see "About Clusters," on page 150 in Chapter 6, "Managing Server Clusters."

`magnus.conf`: the main iPlanet Web Server configuration file. This file contains global server configuration information (such as, port, security, and so on). This file sets the values for variables that configure the server during initialization. iPlanet Web Sever reads this file and executes the variable settings on startup. The server does not read this file again until it is restarted, so you must restart the server every time you make changes to this file. For more information, see "Viewing Server Settings," on page 166 in Chapter 7, "Configuring Server Preferences."

`obj.conf`: the server's object configuration file. This file contains additional initialization information, settings for server customization, and instructions that the server uses to process requests from clients (such as browsers). iPlanet Web Server reads this file every time it processes a client request. For more information, see "Viewing Server Settings," on page 166 in Chapter 7, "Configuring Server Preferences."

For more information about the actual file syntax and the specific directives used by the `obj.conf` and `magnus.conf` configuration files, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

`mime.types`: the MIME (Multi-purpose Internet Mail Extension) type configuration file. This file maps file extensions to MIME types, to enable the server to determine the type of content being requested. For example, requests for resources with `.html` extensions indicate that the client is requesting an

HTML file, while requests for resources with `.gif` extensions indicate that the client is requesting an image file in GIF format. For more information, see "Specifying a Default MIME Type," on page 326 in Chapter 13, "Managing Server Content." Note that you must restart the server every time you make changes to this file.

`admpw`: the username and password file for the Administrator Server superuser. For more information, see "Changing the Superuser Settings," on page 67 in Chapter 3, "Setting Administration Preferences."

Single-Server Configuration

If you have installed iPlanet Web Server on a single server, the installation process places all the files under the server root directory that you specified during installation.

All Platforms

For all platforms, the following directories are created under the server root directory:

- **`alias`** contains the key and certificate files for all Netscape/iPlanet servers (for example, `https-adserv-`*`serverid-cert7.db`* and `secmod.db`).
- **`bin`** contains the binary files for the server, such as the actual server, the Administration Server forms, and so on. In addition, this directory includes the `https/install` folder that contains files needed for migrating server settings and default configuration files needed for backward compatibility.
- **`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 the log analyzer and log analysis tools.
  - 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.

- **`https-admserv`** contains the directories for the Administration Server. This directory has the following subdirectories and files:

    - For Unix/Linux platforms, this directory contains shell scripts to start, stop, and restart the server and a script to rotate log files.

    - `conf_bk` contains backup copies of the server's configuration files.

    - `config` contains the server's configuration files: `admpw, cron.conf, dsgw.conf, dsgwfilter.conf, dsgwlanguage.conf, dsgw-orgperson.conf, dsgwserarchprefs.conf, magnus.conf, magnus.conf.clfilter, mime.types, ns-cron.conf, obj.conf, obj.conf.clfilter, servers.lst.` Working copies are kept here. For more information on `magnus.conf` and `obj.conf`, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

    - `logs` contains any error or access log files.

    - `startsvr.bat` is the script that starts the Server Manager. The Server Manager lets you configure all servers installed in the server root directory.

    - `stopsvr.bat` is the script that stops the Server Manager.

- **`https-server_id.domain`** are the directories for each server you have installed on the machine. Each server directory has the following subdirectories and files:

    - `ClassCache` contains classes and Java files, generated as result of the compilation of JavaServer pages.

  - `conf_bk` contains backup copies of the server's configuration files.
  - `config` contains the Administration Server configuration files.
  - `logs` contains the Administration Server log files.
  - `search` contains the following directories: `admin` and `collections`
  - `SessionData` contains session database data from `MMapSessionManager`.
  - `startsvr.bat` is the script that starts the Server Manager. The Server Manager lets you configure all servers installed in the server root directory.
  - `stopsvr.bat` is the script that stops the Server Manager.
- **`manual`** contains the online manuals for the product.
- **`plugins`** contains directories for Java, search, and other plugins. This directory has the following subdirectories:
  - `content_mgr` contains directories for your server's content.
  - `htaccess` contains server plugin for `.htaccess` access control and `htconvert`, an `.nsconfig` to `.htaccess` converter.
  - `include` contains various include files.
  - `lib` contains shared libraries.
  - `nsacl` contains information for your server's access control lists.
  - `loadbal` contains the required files for the Resonate load-balancer integration plugin.
  - `nsapi` contains header files and example code for creating your own functions using NSAPI. For more information, see the iPlanet documentation web site at: `http://www.iplanet.com/docs`
  - `samples/js` contains the Application Manager and the samples for server-side JavaScript. Note that this is available only if JavaScript was installed.

  - `search` contains information for your server's search plugins.
  - `snmp` contains information for your server's SNMP plugins.
- **`setup`** contains the various iPlanet Web Server setup files.
- **`userdb`** contains user databases and related information.
- **`LICENSE.txt`** is the license file.
- **`README.txt`** is the readme file that contains a link to the iPlanet Web Server, Enterprise Edition 4.1 *Release Notes*.

Unix and Linux Platforms

In addition to the files and directories described in "All Platforms," the following files are created at the `server-root` directory for Unix and Linux platforms:

- **`startconsole`** launches a browser to the Administration Server page.

The following files are created under the **`server-root/https-admserv`** directory for Unix and Linux platforms:

- `ClassCache` contains classes and Java files, generated as result of the compilation of JavaServer pages.
- `conf_bk` contains backup copies of the server's configuration files.
- `config` contains the Administration Server configuration files.
- `logs` contains the Administration Server log files.
- `SessionData` contains session database data from `MMapSessionManager`.
- `restart` is the script that restarts the Server Manager.
- `rotate`
- `start` is the script that starts the Server Manager. The Server Manager lets you configure all servers installed in the server root directory.
- `stop` is the script that stops the Server Manager.

Multiple-Server Configuration

You can also have multiple Web servers running on the same server—all of which can be configured from a single-server administration interface called Administration Server, or from the client-side application, Netscape Console. For more information about Netscape Console, see "Netscape Console."

For more information regarding how to use the Administration Server to configure multiple servers on your machine, see "Setting Encryption Preferences," on page 71 in Chapter 3, "Setting Administration Preferences."

Administration Server

The Administration Server is a web-based server that contains the Java and JavaScript forms you use to configure all of your iPlanet Web Servers.

After installing iPlanet Web Server, you use your browser to navigate to the Administration Server page and use its forms to configure your iPlanet Web Servers. When you submit the forms, the Administration Server modifies the configuration for the server you were administering.

The URL you use to navigate to the Administration Server page depends on the computer host name and the port number you choose for the Administration Server when you install iPlanet Web Server. For example, if you installed the Administration Server on port 1234, the URL would look like this:

```
http://myserver.mozilla.com:1234
```

Before you can get to any forms, the Administration Server prompts you to authenticate yourself. This means you need to type a user name and password. You set up the "superuser" user name and password when you install iPlanet Web Server on your computer. After installation, you can use distributed administration to give multiple people access to different forms in the Administration Server. For more information about distributed administration, see "Enabling Distributed Administration," on page 69 in Chapter 3, "Setting Administration Preferences."

The first page you see when you access the Administration Server, is called Servers. You use the buttons on this page to manage, add, remove, and migrate your iPlanet Web Servers. The Administration Server provides the following tabs for your administration-level tasks:

- Servers
- Preferences
- Global Settings
- Users and Groups
- Security
- Cluster Mgmt (Cluster Management)

Note You must enable cookies in your browser to run the CGI programs necessary for configuring your server.

For more information on using the Administration Server, including information regarding these administration-level tasks, see Chapter 2, "Administering iPlanet Web Servers."

Server Manager

The Server Manager is a web-based interface that contains the Java and JavaScript forms you use to configure individual instances of iPlanet Web Server.

This section includes the following topics:

- Accessing the Server Manager
- Using the Resource Picker
- Wildcards Used in the Resource Picker

Accessing the Server Manager

You can access the Server Manager for iPlanet Web Server by performing the following steps:

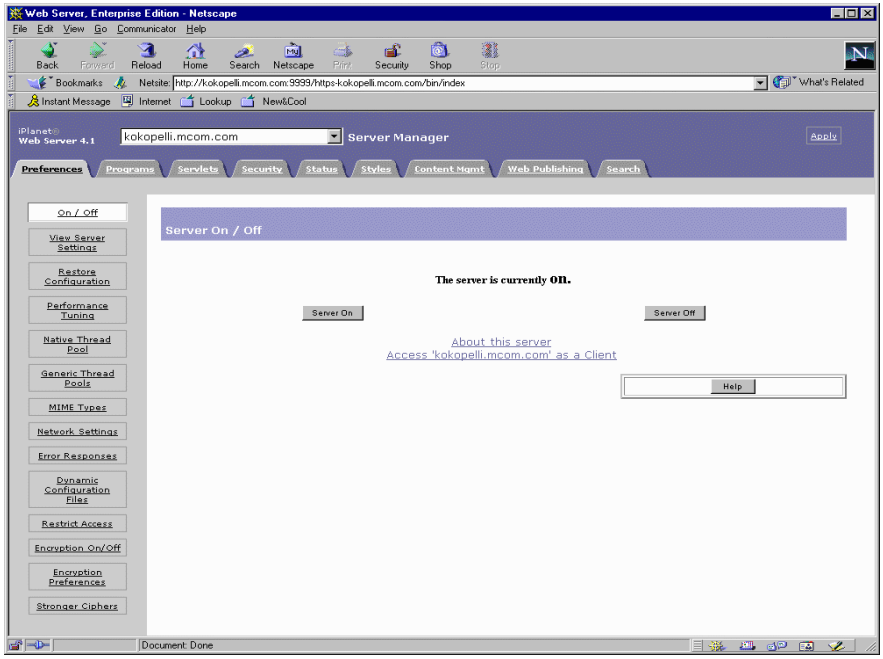
1. Install and start your iPlanet Web Server.

   The Administration Server displays the **Servers** page.

2. In the **Manage Servers** area, select the desired server and click **Manage**.

   iPlanet Web Server displays the Server Manager Preferences page, as shown in the following illustration:

Figure 1.1 The iPlanet Web Server Server Manager



Note Note that you must enable cookies in your browser to run the CGI programs necessary for configuring your server.

You use the links on the Preferences page to manage the following options:

- Turn iPlanet Web Server on/off
- Server settings
- Restore configuration
- Performance tuning actions
- Native thread pool
- Generic thread pool
- Global MIME types
- Network settings
- Error responses
- Dynamic configuration files
- Restrict access
- Encryption preferences
- Stronger ciphers

In addition, the Server Manager provides the following tabs for additional iPlanet Web Server managerial tasks:

- Programs
- Servlets
- Security
- Status
- Styles
- Content Mgmt
- Web Publishing
- Search

For more information, see “Server Manager”, in the online help.

Using the Resource Picker

Most of the Server Manager pages configure the entire iPlanet Web Server. Some pages can configure either the entire server or files or directories that the server maintains. These pages include the **Resource Picker**, shown in Figure 1.2, at the top. The Resource Picker lets you specify what resource to configure.

Figure 1.2 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 1.1 Resource Picker wildcard patterns

Wildcard Pattern	Description
*	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 1.1 Resource Picker wildcard patterns

Wildcard Pattern	Description
`[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.
`*.iplanet.com`	Matches any string ending with the characters `.iplanet.com`.
`quark|energy).iplanet.com`	Matches either `quark.iplanet.com` or `energy.iplanet.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.
`~iplanet-*`	Matches any string except those starting with `iplanet-`.
`*.iplanet.com~ quark.iplanet.com`	Matches any host from domain `iplanet.com` except for a single host `quark.iplanet.com`.
`*.iplanet.com~ (quark|energy|neutrino).iplanet.com`	Matches any host from domain `iplanet.com` except for hosts `quark.iplanet.com`, `energy.iplanet.com`, and `neutrino.iplanet.com`.
`*.com~*.iplanet.com`	Matches any host from domain `com` except for hosts from subdomain `iplanet.com`.

Netscape Console

Netscape Console is a Java application that provides server administrators with a graphical interface for managing all Netscape/iPlanet servers from one central location anywhere within your enterprise network. From any installed instance of Netscape Console, you can see and access all the Netscape/iPlanet

servers on your enterprise's network to which you have been granted access rights. You can log in from any system connected to your network to manage a remote server or to make changes in a centralized directory.

Note For any given instance of Netscape Console, the limits of the network it can administer are defined by the set of resources whose configuration information is stored in the same configuration directory. That is the maximum set of hosts and servers that can appear in the Console window. For a given administrator using Netscape Console, the actual number of visible servers and hosts may be fewer, depending on the access permissions that administrator has.

For complete documentation on Netscape Console, see *Managing Servers with Netscape Console*.

Sending Error Information

iPlanet Web Server includes an error-handling mechanism called the **Quality Feedback Agent**. The Quality Feedback Agent enables you to automatically send error information (stack and register dump) to the Sun-Netscape Alliance if your iPlanet Web Server crashes.

By enabling the Quality Feedback Agent, you can assist the Sun-Netscape Alliance in determining the cause of errors that occur in the server. The Quality Feedback Agent only sends the Sun-Netscape Alliance information to help determine the cause of the error; it does not send documents or other sensitive information.

Details on Data Collected by the Quality Feedback Agent

The Quality Feedback Agent collects only the information needed to analyze and fix errors in the iPlanet Web Server. The following table summarizes all of the information collected by the agent and the reason why the Sun-Netscape Alliance collects this information.

Table 1.2 Data Collected by Quality Feedback Agent

Data Collected	OS-specific Data	Reason for Data Collection
Stack Trace	**Windows & Unix/Linux:** Stack Trace	Shows where iPlanet Web Server failed and what functions were called just before the failure.
PC (Program Counter)	**Windows & Unix/Linux:** PC	Can be used to see if the iPlanet Web Server was in a bad state when it failed.
Registers	**Windows:** Processor Registers **Unix/Linux:** No	Provides the state of the processor at the time of the failure.
Dynamic Libraries	**Windows:** Loaded dlls **Unix/Linux:** ELF32 Shared Objects	Shows any additional dlls that might have been running with or missing from the iPlanet Web Server when it failed.
Threads	**Windows:** Threads in Active Process **Unix/Linux:** No	Identifies potential race conditions with other applications or with different processes in the iPlanet Web Server.
OS Version	**Windows:** Windows Version **Unix/Linux:** Unix Version	Provides the OS version. This information is necessary because the way the iPlanet Web Server interacts with different versions of an OS can cause different kinds of failures.
Processor Type	**Windows:** Processor Information **Unix/Linux:** Processor Information	Provides the processor version. This information is necessary because the iPlanet Web Server, like many software applications, can behave differently when it is running on different-speed processors.
Stack Data	**Windows & Unix/Linux:** Top 2048 bytes on the stack	Shows the value of variables passed into a function that was running at the time of failure.

Using the Quality Feedback Agent

The Quality Feedback Agent enables you to automatically send error information (stack and register dump) to the Sun-Netscape Alliance if your iPlanet Web Server crashes.

By enabling the Quality Feedback Agent, you can assist the Sun-Netscape Alliance in determining the cause of errors that occur in the server. The Quality Feedback Agent only sends the Sun-Netscape Alliance information to help determine the cause of the error; it does not send documents or other sensitive information.

Note If JVM is enabled, you can not use Quality Feedback Agent.

To enable the Quality Feedback Agent for your iPlanet Web Server, perform the following procedures:

1. If necessary, edit your `master.ini` file to allow the Quality Feedback Agent to send data through your firewall to the Sun-Netscape Alliance. For more information, see "Editing master.ini."

2. Edit `magnus.conf` to enable the Quality Feedback Agent (plus any optional parameters) for your iPlanet Web server. For more information, see "Editing magnus.conf."

Editing master.ini

If you are using automatic proxy configuration, and you want to use the Quality Feedback Agent to send incident reports to the Sun-Netscape Alliance, you need to edit the `master.ini` file to contain the appropriate proxy configuration information.

To enable the Quality Feedback Agent, perform the following steps:

1. If you are using an HTTP proxy, or both an HTTP and SOCKS proxy, open the file `master.ini` in the *`server_root`*`/bin/https/bin` directory.

2. Add the following three lines of code to your `master.ini` file, using your proxy host name, domain, and port:

```
UseUserHTTPProxyInfo=1
```

```
UserHTTPProxyHost="yourproxy.yourdomain.com"

UserHTTPProxyPort=xxxx
```

If you are using a SOCKS Proxy, add the following three lines of code to your `master.ini` file:

```
UseUserSOCKSInfo=1

UserSOCKSHost="yourproxy.yourdomain.com"

UserSOCKSPort=xxxx
```

Editing magnus.conf

To turn on the Quality Feedback Agent for your iPlanet Web server, add TalkBack on to your `magnus.conf` file. To disable it, either delete TalkBack, or specify TalkBack off.

In addition, there are two optional `magnus.conf` file variables for the Quality Feedback Agent:

- `TalkbackMaxIncidents`: If the server crashes more often than this number within a time interval, the Quality Feedback Agent will be turned off automatically. The default is 5.
- `TalkbackInterval`: The interval used by the parameter above, in seconds. The default is 86400 seconds (24 hours).

Note that both variables have no effect unless the Quality Feedback Agent is turned on. Once you restart the server, the counters are reset and the whole process starts over.

Chapter

2

Administering iPlanet Web Servers

This chapter describes how to administer your iPlanet Web Servers with the iPlanet Web Server Administration Server. Using the Administration Server, you can manage servers, add and remove servers, and migrate servers from a previous release.

This chapter includes the following sections:

- Accessing the Administration Server
- Adding a Server: Running Multiple Servers
- Installing Multiple Instances of the Server
- Removing a Server
- Migrating a Server From a Previous Version

Accessing the Administration Server

This section describes how to access the Administration Server for Unix/Linux and Windows NT platforms.

Unix/Linux Platforms

To access the Administration Server in Unix or Linux platforms, go to the *server_root*/https-admserv/ directory (for example, /usr/netscape/server4/https-admserv/) and type ./start. This command starts the Administration Server using the port number you specified during installation.

Windows NT Platforms

The iPlanet Web Server installation program creates a program group with several icons for Windows NT platforms. The program group includes the following icons:

- Release Notes
- Start Administration Server
- Uninstall iPlanet Web Server 4.1

Note that the Administration Server runs as a services applet; thus, you can also use the Control Panel to start this service directly.

To access the Administration Server in Windows NT 4.0, perform the following steps:

1. Double-click the "Start Administration Server" icon, or type the following URL in your browser:

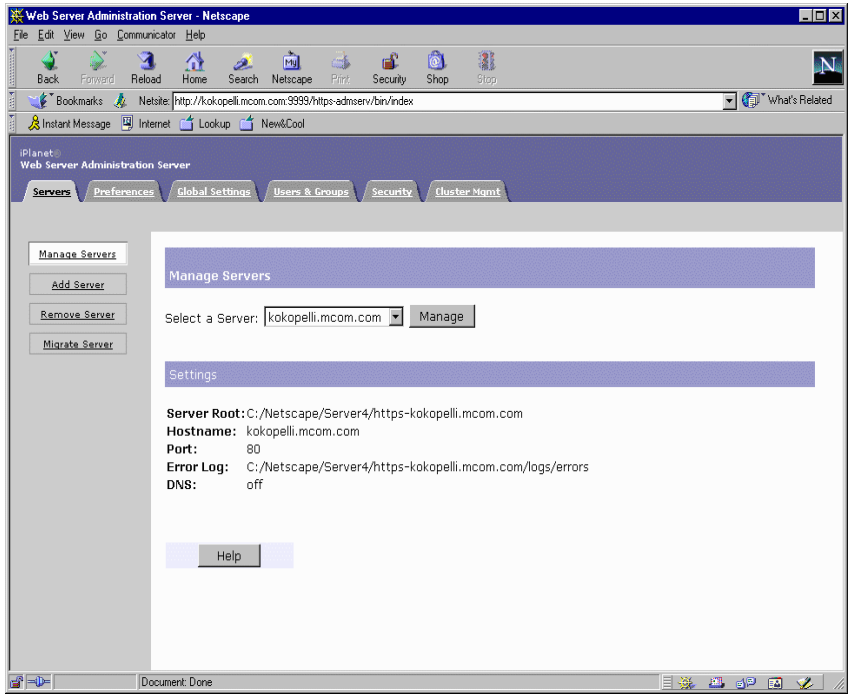
   http://*hostname.domain-name:administration_port*

   iPlanet Web Server then displays a window prompting you for a username and password.

2. Type the administration username and password you specified during installation.

   iPlanet Web Server displays the Administration Server page, as shown in Figure 2.1:

Figure 2.1 The Administration Server Page



For more information, see "Administration Server," in the online help.

Note You must enable cookies in your browser to run the CGI programs necessary for configuring your server.

You can also access the Administration Server from a remote location as long as you have access to client software such as Netscape Navigator. Since the Administrator Server is accessed through a browser, you can access it from any machine that can reach the server over the network. For more information, see "Netscape Console," on page 50 in Chapter 1, "Introduction to iPlanet Web Server."

Adding a Server: Running Multiple Servers

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

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

Hardware Virtual Servers

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.

Configuring Multiple Hardware Virtual Servers on the Same IP Address with Different Ports

You can configure multiple hardware virtual servers on the same IP address by assigning a unique port number for each hardware virtual server. Note that you can not use the Administration Server to accomplish this task, you must manually edit the `obj.conf` file.

For example, to configure four hardware virtual servers on the same IP address with port numbers 80-83, add the following lines to the `obj.conf` file:

```
NameTrans fn="document-root" dir="/vs1docroot"
ip="a.b.c.d" port="80"

NameTrans fn="document-root" dir="/vs2docroot"
ip="a.b.c.d" port="81"

NameTrans fn="document-root" dir="/vs3docroot"
ip="a.b.c.d" port="82"
```

```
NameTrans fn="document-root" dir="/vs4docroot"
ip="a.b.c.d" port="83"
```

For more information on hardware virtual servers, see "Setting Up Hardware Virtual Servers," on page 327 in Chapter 13, "Managing Server Content."

Software Virtual Servers

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 software virtual servers, see "Setting up Software Virtual Servers," on page 331 in Chapter 13, "Managing Server Content."

Multiple Server Instances

Multiple server instances enables you to define separate types of configuration information for each server. 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. For more information, see the following section, "Installing Multiple Instances of the Server."

Installing Multiple Instances of the Server

You can use the Administration Server to configure multiple servers via the following options:

- Install multiple copies of the server on NT as separate instances, each with a different IP address.

- Configure a number of additional hardware virtual servers, with one iPlanet Web Server which responds to the various virtual servers independently.
- Configure a number of software virtual servers, which enables you to host multiple web sites from one IP address.
- Configure a set of servers that all use the same IP address, but different port numbers.

If you have installed iPlanet Web Server on multiple servers, the installation process places all the files under the server root directory that you specified during installation, as specified in "Single-Server Configuration," in Chapter 1, "Introduction to iPlanet Web Server." However, note that iPlanet Web Server also creates an additional `https-identifier` directory for each additional server you specify.

You can install another instance of the web server on your current computer. 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 and install separate instances of the server for each IP address. For more information, see "Configuring Network Settings," on page 169 in Chapter 7, "Configuring Server Preferences."

To add another server instance, perform the following steps:

1. Access the Administration Server and choose the **Servers** tab.
2. Click the **Add Server** link.
3. Enter the desired information for the specified fields.

For more information, see "The Add Server Page," in the online help.

Removing a Server

You can remove a server from your system using the Administration Server. Be sure that you don't need the server anymore before you remove it, since this process cannot be undone.

Note Some NT servers have an uninstall program that you can use to remove a server and its associated administration server. For details, check with your product documentation.

To remove a server from your machine, perform the following steps:

1. Access the Administration Server and choose the **Servers** tab.
2. Click **Remove Server**.

The Administration Server subsequently deletes the server's configuration files, Server Manager forms, and the following directory (and any subdirectories):

server_root/<servertype>-<id>

For more information, see "The Remove Server Page," in the online help.

Migrating a Server From a Previous Version

You can migrate an Enterprise Server from 3.6 to 4.1. Your 3.6 server is preserved, and a new 4.1 server using the same settings is created.

You should stop running the 3.6 server before migrating settings. Make sure you have Netscape Navigator 3.0 or later installed on your computer before migrating settings.

For a complete description of how to migrate a server from a previous version to Enterprise Server 4.1, see the *Installation and Migration Guide*.

For more information, see "The Migrate Server Page," in the online help.

2

Using the Administration Server

- **Setting Administration Preferences**
- **Managing Users and Groups**
- **Working with Server Security**
- **Managing Server Clusters**

Chapter

3

Setting Administration Preferences

This document describes the administration forms available via the Preferences and Global Settings tabs in the Administration Server that you use to configure your iPlanet Web Servers. Note that you must enable cookies in your browser to run the CGI programs necessary for configuring your server.

This chapter includes the following sections:

- Shutting Down the Administration Server
- Changing Network Settings
- Changing the Superuser Settings
- Enabling Distributed Administration
- Configuring Secure Sockets Layer (SSL)
- Specifying Log File Options
- Configuring Directory Services
- Restricting Server Access

Shutting Down the Administration Server

Once the server is installed, it runs constantly, listening for and accepting HTTP requests. You can stop the server using one of the following methods:

- Access the Administration Server, choose the Servers tab, and perform the following steps:

  1. Select the **Manage Servers** option.

  2. Select the server you want to shut down from the Select a Server drop-down list.

  3. Click **Manage**. The iPlanet Web Server displays the Server Manager forms.

  For more information about using the Server On/Off page, see "Starting and Stopping the Server," on page 160 in Chapter 7, "Configuring Server Preferences."

- Choose the **Preferences** tab, select the **Shut Down** option, and click **Shut down the administration server!** button. For more information, see "The Shutdown Page," in the online help.

- Use the Services window in the Control Panel (Windows NT).

- Use `stop`, which shuts down the server completely, interrupting service until it is restarted. If you set the `etc/inittab` file to automatically restart (using "`respawn`"), you must remove the line pertaining to the web server in `etc/inittab` before shutting down the server; otherwise, the server automatically restarts. (Unix/Linux platforms).

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."

Changing Network Settings

Network settings affect the way the Administration Server works with your iPlanet Web Servers. You can change the system user account and password and port number for iPlanet Web Administration Server.

Changing the User Account and Password

To change the system user account, you must use the Server Manager forms. For more information, see “Configuring Network Settings,” on page 169 in Chapter 7, “Configuring Server Preferences.”

NT You can also change the password that the server uses when the service starts. Make sure that the user account has a password and has both administrative and “log on as a service” permissions. You should change the permissions using the Windows NT User Manager program located in the Administrative Tools group for your desktop.

Changing the Port Number

You can also change the port number that the Administration Server listens to. The port number can be any number between 1 and 65535, but it is typically a random number greater than 1024. For security reasons, consider changing the port number regularly.

To change the Administration Server port number, perform the following steps:

1. Access the Administration Server and choose the **Preferences** tab.
2. Click the **Network Settings** link.
3. Make the desired changes and click OK.

Note that you must restart the server for the settings to take effect.

For more information, see “The Network Settings Page,” in the online help.

Changing the Superuser Settings

You can configure superuser access for your Administration Server. These settings affect only the superuser account. That is, if your Administration Server uses distributed administration, you need to set up additional access controls for the administrators you allow.

Warning If you use Netscape Directory Server to manage users and groups, you need to update the superuser entry in the directory *before* you change the superuser username or password. If you don't update the directory first, you won't be able to access the Users & Groups forms in the Administration Server. To fix this, you'll need to either access the Administration Server with an administrator account that does have access to the directory, or you'll need to update the directory using the Netscape Directory Server's Netscape Console or configuration files.

To change the superuser settings for the Administration Server, perform the following steps:

1. Access the Administration Server and choose the **Preferences** tab.
2. Click the **Superuser Access Control** link.
3. Make the desired changes and click OK.

For more information, see "The Superuser Access Control Page," in the online help.

Note You can change the Administration Server user from root to another user on the operating system to enable multiple users (belonging to the group) to edit/manage the configuration files. However, note that while on Unix/Linux platforms, the installer can give "rw" (read/write) permissions to a group for the configuration files, on Windows NT platforms, the user must belong to the "Administrators" group.

The superuser's username and password are kept in a file called *server_root*`/admin-serv/config/admpw`. If you forget the username, you can view this file to obtain the actual name; however, note that the password is encrypted and unreadable. The file has the format `username:password`.

Warning If you forget the password, you can edit the `admpw` file and simply delete the encrypted password. You can then go to the Server Manager forms and specify a new password. Because you can do this, it is very important that you keep the server computer in a secure place and restrict access to its file system. On Unix/Linux systems, consider changing the file ownership so that it's writable only by root or whatever system user runs the Administration Server daemon. On NT systems, restrict the file ownership to the user account Administration Server uses.

Enabling Distributed Administration

Distributed administration allows multiple administrators to change specific parts of the server. With distributed administration you have three levels of users:

- **superuser** is the user listed in the file *server_root*/admin-serv/config/admpw. This is the user name (and password) you specified during installation. This user has full access to all forms in the Administration Server, except the Users & Groups forms, which depend on the superuser having a valid account in an LDAP server such as Netscape Directory Server.
- **administrators** go directly to the Server Manager forms for a specific server, including the Administration Server. The forms they see depend on the access control rules set up for them (usually done by the superuser). Administrators can perform limited administrative tasks and can make changes that affect other users, such as adding users or changing access control.
- **end users** can view read-only data stored in the database. Additionally, end users may be granted access permissions to change only specific data.

For an in-depth discussion of access control for iPlanet Web Server, see "What Is Access Control?," on page 336 in Chapter 14, "Controlling Access to Your Server."

Note Before you can enable distributed administration, you must install a Directory Server. For more information, see *Netscape Directory Server Administrator's Guide*.

To enable distributed administration, perform the following steps:

1. Verify that you have installed a Directory Server.
2. Access the Administration Server.
3. One you've installed a Directory Server, you may also need to create an administration group, if you have not previously done so.

To create a group, perform the following steps:

1. Choose the **Users & Groups** tab.
2. Click the **New Group** link.
3. Create an "administrators" group in the LDAP directory and add the names of the users you want to have permission to configure the Administration Server, or any of the servers installed in its server root. All users in the "administrators" group have full access to the Administration Server, but you can use access control to limit the servers and forms they will be allowed to configure.

Warning Once you create an access-control list, the distributed administration group is added to that list. If you change the name of the "administrators" group, you must manually edit the access-control list to change the group it references.

4. Choose the **Preferences** tab.
5. Click the **Distributed Admin** link.
6. Make the desired changes and click OK.

For more information, see "The Distributed Administration Page," in the online help.

Configuring Secure Sockets Layer (SSL)

Using the Administration Server, you can activate the iPlanet Web Server encryption feature and set various encryption preferences. For more information regarding iPlanet Web Server encryption features, see "About iPlanet Web Server Security," on page 110 in Chapter 5, "Working with Server Security."

Note that prior to activating SSL for your iPlanet Web Server you need to set up some preliminary requirements, such as creating a trust database, and requesting and installing an encryption certificate. For more information, see "Configuring iPlanet Web Server for SSL," on page 116 in Chapter 5, "Working with Server Security."

Activating SSL

To activate SSL for your Administration Server, perform the following steps:

1. Access the Administration Server and choose the **Preferences** tab.
2. Click the **Encryption On/Off** link.
3. Make the desired changes and click OK.

For more information, see "The Encryption On/Off Page," in the online help.

Setting Encryption Preferences

The Administration Server enables you to set the following SSL encryption preferences:

- Choose between various versions of SSL.
- Specify whether to require client certificates.
- Set the SSL 2.0 ciphers.
- Set the SSL 3.0 ciphers.

Your server can perform encryption with a number of different encryption functions, called **ciphers**. Some ciphers are more resistant to cracking than others. During an SSL connection, the client and the server agree to use the strongest cipher they can both use for communication. For more information regarding ciphers, see *Managing Servers with Netscape Console*.

To set these encryption preferences, perform the following steps:

1. Access the Administration Server and choose the **Preferences** tab.
2. Click the **Encryption Prefs** link.
3. Check the SSL versions you want your server to communicate with. The latest and most secure version is SSL version 3, but a few older clients use only SSL version 2. You will probably want to enable your server to use both versions.

4. Check the ciphers you want your server to use. The ciphers are listed for each version of SSL. 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. Ciphers are described after this list.

5. Click OK. Make sure you restart your server.

When a client initiates an SSL connection with a server, the client lets the server know what ciphers it prefers to use to encrypt information. In any two-way encryption process, both parties must use the same ciphers. Since there are a number of ciphers available, you should consider enabling all ciphers.

You can choose ciphers from both the SSL 2 and SSL 3 protocols. Unless you have a compelling reason why you don't want to use a specific cipher, you should check them all.

For more information, see "The Encryption Preferences Page," in the online help.

Setting Stronger Ciphers

You can set stronger ciphers via the **Stronger Ciphers** option on the Server Manager **Preferences** tab. The Stronger Ciphers option presents a choice of 168, 128, or 56-bit secret keysize restriction, or no restriction. You can specify a filename to be served when the restriction is not met. If no filename is specified, iPlanet Web Server returns a "Forbidden" status.

If you select a restriction that is not consistent with the current cipher settings under Security Preferences, iPlanet Web Server displays a popup dialog that warns that you need to enable ciphers with larger secret keysizes.

The implementation of the keysize restriction is now based on an NSAPI `PathCheck` directive, rather than Service `fn=key-toosmall`. This directive is:

```
PathCheck fn="ssl-check" [secret-keysize=<nbits>]
[bong-file=<filename>]
```

where `<nbits>` is the minimum number of bits required in the secret key, and `<filename>` is the name of a file (not a URI) to be served if the restriction is not met.

This function returns `REQ_NOACTION` if SSL is not enabled, or if the `secret-keysize` parameter is not specified. If the secret keysize for the current session is less than the specified `secret-keysize`, the function returns `REQ_ABORTED` with a status of `PROTOCOL_FORBIDDEN` if `bong-file` is not specified, or else `REQ_PROCEED`, and the "path" variable is set to the `bong-file <filename>`. Also, when a keysize restriction is not met, the SSL session cache entry for the current session is invalidated, so that a full SSL handshake will occur the next time the same client connects to the server.

Note The Stronger Ciphers form removes any Service `fn=key-toosmall` directives that it finds in an object when it adds a PathCheck `fn=ssl-check`.

For more information, see "The Enforce Strong Security Requirements Page," in the online help.

Specifying Log File Options

Log files can help you monitor your server's activity. You can use these logs to monitor your server and troubleshoot problems.

To configure logging options for the Administration Server, perform the following steps:

1. Access the Administration Server and choose the **Preferences** tab.
2. Click the **Logging Options** link.
3. Make the desired changes and click OK.

For more information, see "The Logging Options Page," in the online help.

This section also includes topics that describe how to configure the iPlanet Web Server Log File options to perform the following tasks:

- Viewing the Access Log File
- Viewing the Error Log File
- Archiving Log Files

Viewing the Access Log File

The `access` log, located in `admin/logs` in the server root directory, records information about requests to the server and the responses from the server.You can specify the server log format—what is included in the `access` log file—to be the Common Logfile Format, a commonly supported format that provides a fixed amount of information about the server, or you can create a custom log file format that better suits your server requirements.

To view the access log file, perform the following steps:

1. Access the Administration Server and choose the **Preferences** tab.
2. Click the **View Access Log** link and click OK.

For more information, see "The View Error Log Page," in the online help.

Viewing the Error Log File

The `error` log file, located in `admin/logs` in the server root directory, lists all the 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 and who tried unsuccessfully to log in to the server.

To view the error log file, perform the following steps:

1. Access the Administration Server and choose the **Preferences** tab.
2. Click the **View Error Log** link and click OK.

You can also view the server's active and archived log files from the Server Manager. For more information regarding these log files, see "The View Access Log Page," in the online help.

Archiving Log Files

You can set up your log files to be automatically archived. At a certain time, or after a specified interval, iPlanet Web Server rotates your access logs. iPlanet Web Server saves the old log files and stamps the saved file with a name that includes the date and time they were saved.

For example, you can set up your files to rotate every hour, and iPlanet Web Server saves and names the file "`access.199907152400`," where "name|year|month|day|24-hour time" is concatenated together into a single character string. The exact format of the access log archive file varies depending upon which type of log rotation you set up.

iPlanet Web Server offers the two types of log rotation for archiving files:

- **Internal-daemon log rotation**—this type of log rotation happens within the HTTP daemon, so the server doesn't need to restart.
- **Cron-based log rotation**—this type of log rotation is based on the time stored in the `cron.conf` file. For more information about cron controls, see "Using Cron Controls (Unix/Linux)," on page 75.

Access log rotation is initialized at server startup. If rotation is turned on, iPlanet Web Server creates a time-stamped access log file and rotation starts at server startup.

Once the rotation starts, iPlanet Web Server creates a new time stamped access log file when there is a request that needs to be logged to the access log file and it occurs after the previously-scheduled "next rotate time."

For more information about archiving log files, see "Archiving Log Files," on page 189 in Chapter 8, "Understanding Log Files."

Using Cron Controls (Unix/Linux)

You can configure several features of your iPlanet Web Server to operate automatically and set to begin at specific times. The Netscape cron daemon checks the computer clock and then spawns processes at certain times. (These settings are stored in the `ns-cron.conf` file.)

This cron daemon controls scheduled tasks for your iPlanet Web Server and can be activated and deactivated from the Administration Server. The tasks performed by the cron process depends on the various servers. (Note that on NT platforms, the scheduling occurs within the individual servers.)

Some of the tasks that can be controlled by cron daemons include scheduling collection maintenance and archiving log files. You need to restart cron control whenever you change the settings for scheduled tasks.

To restart, start, or stop cron control, perform the following steps:

1. Access the Administration Server and choose the **Global Settings** tab.
2. Click the **Cron Control** link.
3. Click Restart, Start, or Stop to change the cron controls.

Note that any time you add a task to cron, you need to restart the daemon.

Configuring Directory Services

You can manage all your user information from a single source via an open-systems server protocol called the **Lightweight Directory Access Protocol (LDAP)**. You can also configure the server to allow your users to retrieve directory information from multiple, easily accessible network locations.

To configure the directory services preferences, perform the following steps:

1. Access the Administration Server and choose the **Global Settings** tab.
2. Click the **Configure Directory Service** link.
3. Make the desired changes and click OK.

For more information, see "The Configure Directory Service Page," in the online help.

Restricting Server Access

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.

You can set access control globally for all servers through the Administration Server or for a resource within a specific server instance through the Server Manager. For more information about setting access control for a resource, see "Restricting Access to Your Web Site," on page 344 in Chapter 14, "Controlling Access to Your Server."

Note You must turn on distributed administration before you can restrict server access.

To restrict access to your iPlanet Web Servers, perform the following steps:

1. Access the Administration Server and choose the **Global Settings** tab.
2. Click the **Restrict Access** link.
3. Select the desired server and click **Edit ACL**.

   The Administration Server displays the access control rules for the server you specified.
4. Make the desired access control changes and click OK.

For more information, see "The Restrict Access Page," in the online help.

Chapter

4

Managing Users and Groups

This chapter describes how to use the forms in the Administration Server Users and Groups tab.

This chapter includes the following sections:

- About Users and Groups
- Creating Users
- Managing Users
- Creating Groups
- Managing Groups
- Creating Organizational Units
- Managing Organizational Units
- Managing a Preferred Language List

About Users and Groups

The Administration Server provides you access to your application data about user accounts, group lists, access privileges, organization units, and other user/group-specific information. You can use the Administration Server to create, locate, and manage records for users and groups within your iPlanet Web Servers.

iPlanet Web Server 4.x does not support local LDAP. In order to add users and groups, you must have a directory server installed, such as Netscape Directory Server. If you need to create, locate, or manage records for users and groups on any other servers within your network, you should use Netscape Console with your Directory Server. For more information, see *Managing Servers with Netscape Console*.

Warning (NT) You cannot install Netscape Directory Server 4.0 and iPlanet Web Server 4.x on the same Windows NT machine because of system library conflicts. Install Directory Server on a separate machine and use the Administration Server's Global Settings tab to configure iPlanet Web Server to use that Directory Server.

The Users and Groups tab of the Administration Server enables you to create or modify users, groups, and organizational units. Each user and group in your enterprise is represented by a **Distinguished Name (DN)** attribute. A DN attribute is a text string that contains identifying information for an associated user, group, or object. You use DNs whenever you make changes to a user or group directory entry. For more information regarding distinguished name syntax and frequently used attributes, see *Managing Servers with Netscape Console*.

Note that if you do not currently have a directory, or if you want to add a new subtree to an existing directory, you can use the Directory Server's Administration Server LDIF import function. This function accepts a file containing LDIF and attempts to build a directory or a new subtree from the LDIF entries. You can also export your current directory to LDIF using the Directory Server's LDIF export function. This function creates an LDIF-formatted file that represents your directory. For more information, see your Directory Server documentation.

Creating Users

Use the Users and Groups tab of the Administration Server to create or modify user entries. A user entry contains information about an individual person or object in the database.

This section includes the following topics:

- Guidelines for Creating User Entries
- How to Create a New User Entry
- Directory Server User Entries

Guidelines for Creating User Entries

Consider the following guidelines when using the administrator forms to create new user entries:

- If you enter a given name (or first name) and a surname, then the form automatically fills in the user's full name and user ID for you. The user ID is generated as the first initial of the user's first name followed by the user's last name. For example, if the user's name is Billie Holiday, then the user ID is automatically set to *bholiday*. You can replace this user ID with an ID of your own choosing if you wish.

- The user ID must be unique. The Administration Server ensures that the user ID is unique by searching the entire directory from the search base (base DN) down to see if the user ID is in use. Be aware, however, that if you use the Directory Server `ldapmodify` command line utility (if available) to create a user, that it does not ensure unique user IDs. If duplicate user IDs exist in your directory, the affected users will not be able to authenticate to the directory.

- Note that the base DN specifies the distinguished name where directory lookups will occur by default, and where all iPlanet Web Administration Server's entries are placed in your directory tree. A "DN" is the string representation for the name of an entry in a directory server.

- Note that at a minimum, you must specify the following user information when creating a new user entry:

    - surname or last name

  - full name
  - user ID

- If any organizational units have been defined for your directory, you can specify where you want the new user to be placed using the Add New User To list. The default location is your directory's base DN (or root point).

Note The user edit text fields for international information differs between the Administration Server and Netscape Console. In Netscape Console, in addition to the untagged `cn` fields, there is a preferred language `cn` field which doesn't exist in the Administration Server.

How to Create a New User Entry

To create a user entry, read the guidelines outlined in "Guidelines for Creating User Entries," on page 81,.and then perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **New User** link and add the associated information to the displayed page.

For more information, see "The New User Page," in the online help. For information on editing users, see "Managing Users," on page 84.

Directory Server User Entries

The following user entry notes may be of interest to the directory administrator:

- User entries use the `inetOrgPerson`, `organizationalPerson`, and `person` object classes.
- By default, the distinguished name for users is of the form:

  ```
  cn=full name, ou=organization, ...,o=base
  organization, c=country
  ```

  For example, if a user entry for Billie Holiday is created within the organizational unit Marketing, and the directory's base DN is o=Ace Industry, c=US, then the person's DN is:

```
cn=Billie Holiday, ou=Marketing, o=Ace Industry, c=US
```

However, note that you can change this format to a uid-based distinguished name.

- The values on the user form fields are stored as the following LDAP attributes (note that any stored information other than 'user' and 'group' requires a full Directory Server license):

Table 4.1 LDAP Attributes

User Field	Corresponding LDAP Attribute
Given Name	`givenName`
Surname	`sn`
Full Name	`cn`
User ID	`uid`
Password	`userPassword`
Email Address	`mail`

The following fields are also available when editing the user entry:

Table 4.2 User Entry LDAP Attributes

User Field	Corresponding LDAP Attribute
Title	`title`
Telephone	`telephoneNumber`

- Sometimes a user's name can be more accurately represented in characters of a language other than the default language. You can select a preferred language for users so that their names will display in the characters of the that language, even when the default language is English. For more information regarding setting a user's preferred language, see "The Manage Users Page," in the online help.

Managing Users

You edit user attributes from the Administration Server Manage Users form. From this form you can find, change, rename, and delete user entries; manage user licenses; and potentially change product-specific information.

Some, but not all, Netscape/iPlanet servers add additional forms to this area that allow you to manage product-specific information. For example, if a messaging server is installed under your Administration Server, then an additional form is added that allows you to edit messaging server-specific information. See the server documentation for details on these additional management capabilities.

This section includes the following topics:

- Finding User Information
- Editing User Information
- Managing a User's Password
- Managing User Licenses
- Renaming Users
- Removing Users

Finding User Information

Before you can edit a user entry, you must display the associated information. To find the specific user information, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Users** link.
3. In the **Find User** field, enter some descriptive value for the entry that you want to edit. You can enter any of the following in the search field:
    - A name. Enter a full name or a partial name. All entries that equally match the search string will be returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
    - A user ID.

- A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number will be returned.
- An email address. Any search string containing an at (@) symbol is assumed to be an email address. If an exact match cannot be found, then a search is performed to find all email addresses that begin with the search string.
- An asterisk (*) to see all of the entries currently in your directory. You can achieve the same effect by simply leaving the field blank.
- Any LDAP search filter. Any string that contains an equal sign (=) is considered a search filter.

As an alternative, use the pull down menus in the Find all users whose field to narrow the results of your search.

4. In the **Look within** field, select the organizational unit under which you want to search for entries. The default is the directory's root point (or top most entry).
5. In the **Format** field, choose either On-Screen or Printer.
6. Click **Find**. All the users in the selected organizational unit are displayed.
7. In the resulting table, click the name of the entry that you want to edit.
8. The user edit form is displayed. Change the displayed fields as desired and click Save Changes. The changes are made immediately.

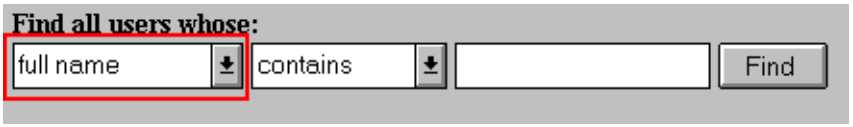
Building Custom Search Queries

The Find all users whose field allows you to build a custom search filter. Use this field to narrow down the search results returned by a "Find user" search.

The Find all users whose field provides the following search criteria:

- The left-most pull-down list allows you to specify the attribute on which the search will be based, as shown in the following illustration:

Figure 4.1 Search Attribute



For a complete list of the available search attribute options, see "Search Attribute Options."

- In the center pull-down list, select the type of search you want to perform, as shown in the following illustration:

Figure 4.2 Search Type

For a complete list of the available search type options, see "Search Type Options."

- In the right-most text field, enter your search string:

Figure 4.3 Search String

To display all of the users entries contained in the Look Within directory, enter either an asterisk (*) or simply leave this text field blank.

Search Attribute Options

The available search attribute options are described in the following table:

Table 4.3 Search Attribute Options

Option Name	Description
full name	Search each entry's full name for a match.
last name	Search each entry's last name, or surname for a match.
user id	Search each entry's user id for a match.
phone number	Search each entry's phone number for a match.
email address	Search each entry's email address for a match.
unit name	Search each entry's name for a match.
description	Search each organizational unit entry's description for a match.

Search Type Options

The available search type options are described in the following table:

Table 4.4 Search Type Options

Option Name	Description
`contains`	Causes a substring search to be performed. Entries with attribute values containing the specified search string are returned. For example, if you know an user's name probably contains the word "Dylan," use this option with the search string "Dylan" to find the user's entry.
`is`	Causes an exact match to be found. That is, this option specifies an equality search. Use this option when you know the exact value of an user's attribute. For example, if you know the exact spelling of the user's name, use this option.

Table 4.4 Search Type Options

Option Name	Description
`isn't`	Returns all the entries whose attribute value does not exactly match the search string. That is, if you want to find all the users in the directory whose name is not "John Smith," use this option. Be aware, however, that use of this option can cause an extremely large number of entries to be returned to you.
`sounds like`	Causes an approximate, or phonetic, search to be performed. Use this option if you know an attribute's value, but you are unsure of the spelling. For example, if you are not sure if a user's name is spelled "Sarret," "Sarette," or "Sarett," use this option.
`starts with`	Causes a substring search to be performed. Returns all the entries whose attribute value starts with the specified search string. For example, if you know a user's name starts with "Miles," but you do not know the rest of the name, use this option.
`ends with`	Causes a substring search to be performed. Returns all the entries whose attribute value ends with the specified search string. For example, if you know a user's name ends with "Dimaggio," but you do not know the rest of the name, use this option.

Editing User Information

To change a user's entry, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Display the user entry as described in "Finding User Information," on page 84.
3. Edit the field corresponding to the attribute that you wish to change.

For more information, see "The Edit Users Page," in the online help.

Note It is possible that you will want to change an attribute value that is not displayed by the edit user form. In this situation, use the Directory Server `ldapmodify` command line utility, if available.

In addition, note that you can change the user's first, last, and full name field from this form, but to fully rename the entry (including the entry's distinguished name), you need to use the Rename User form. For more information on how to rename an entry, see "Renaming Users," on page 90.

Managing a User's Password

The password you set for user entries is used by the various servers for user authentication.

To change or create a user's password, perform the following steps:

1. Access the Administration Server and choose **Users & Groups** tab.
2. Display the user entry as described in "Finding User Information," on page 84.
3. Make the desired changes and click OK.

For more information, see "The Manage Users Page," in the online help.

Note You can change the Administration Server user from root to another user on the operating system to enable multiple users (belonging to the group) to edit/manage the configuration files. However, note that while on Unix/Linux platforms, the installer can give "rw" permissions to a group for the configuration files, on Windows NT platforms, the user must belong to the "Administrators" group.

You can also disable the user's password by clicking the Disable Password button. Doing this prevents the user from logging into a server without deleting the user's directory entry. You can allow access for the user again by using the Password Management Form to enter a new password.

Managing User Licenses

Administration Server enables you to track which iPlanet server products your users are licensed to use.

To manage the licenses available to the user, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Display the user entry as described in "Finding User Information," on page 84.
3. Click the **Licenses** link at the top of the User Edit form.
4. Make the desired changes and click OK.

For more information, see "The Manage Users Page," in the online help.

Renaming Users

The rename feature changes only the user's name; all other fields are left intact. In addition, the user's old name is still preserved so searches against the old name will still find the new entry.

When you rename a user entry, you can only change the user's name; you cannot use the rename feature to move the entry from one organizational unit to another. For example, suppose you have organizational units for Marketing and Accounting and an entry named "Billie Holiday" under the Marketing organizational unit. You can rename the entry from `Billie Holiday` to `Doc Holiday`, but you cannot rename the entry such that `Billie Holiday` under the Marketing organizational unit becomes `Billie Holiday` under the Accounting organizational unit.

To rename a user entry, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Display the user entry as described in "Finding User Information," on page 84.

Note that if you are using common name-based DNs, specify the user's full name. If you are using uid-based distinguished names, enter the new uid value that you want to use for the entry.

3. Click the **Rename User** button.

4. Change the Given Name, Surname, Full Name, or UID fields as is appropriate to match the new distinguished name for the entry.

5. You can specify that the Administration Server no longer retains the old full name or uid values when you rename the entry by setting the `keepOldValueWhenRenaming` parameter to false. You can find this parameter in the following file:

   server_root`/admin-serv/config/dsgw-orgperson.conf`

For more information, see "The Manage Users Page," in the online help.

Removing Users

To delete a user entry, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.

2. Display the user entry as described in "Finding User Information," on page 84.

3. Click **Delete User**.

For more information, see "The Manage Users Page," in the online help.

Creating Groups

A group is an object that describes a set of objects in an LDAP database. An iPlanet Web Server group consists of users who share a common attribute. There are two ways to define membership of a group: statically and dynamically. **Static groups** enumerate their member objects explicitly. A static

group is a CN and contains `uniqueMembers` and/or `memberURLs` and/or `memberCertDescriptions`. For static groups, the members do not share a common attribute except for the `CN=<Groupname>` attribute.

Dynamic groups allow you to use a LDAP URL to define a set of rules that match only for group members. For Dynamic Groups, the members do share a common attribute or set of attributes that are defined in the `memberURL` filter. For example, if you need a group that contains all employees in Sales, and they are already in the LDAP database under "`ou=Sales,o=Airius.com`," you'd define a dynamic group with the following `memberurl`:

```
ldap:///ou=Sales,o=Netscape??sub?(uid=*)
```

This group would subsequently contain all objects that have an `uid` attribute in the tree below the "`ou=Sales,o=Netscape`" point; thus, all the Sales members.

For static and dynamic groups, members can share a common attribute from a certificate if you use the `memberCertDescription`. Note that these will only work if the ACL uses the SSL method.

Once you create a new group, you can add users, or members, to it.

This section includes the following topics for creating groups:

- Static Groups
- Dynamic Groups

Static Groups

The Administration Server enables you to create a static group by specifying the same group attribute in the DNs of any number of users. A static group doesn't change unless you add a user to it or delete a user from it.

Guidelines for Creating Static Groups

Consider the following guidelines when using the Administration Server forms to create new static groups:

- Static groups can contain other static or dynamic groups.
- You can optionally also add a description for the new group.

- If any organizational units have been defined for your directory, you can specify where you want the new group to be placed using the Add New Group To list. The default location is your directory's root point, or top-most entry.
- When you are finished entering the desired information, click Create Group to add the group and immediately return to the New Group form. Alternatively, click Create and Edit Group to add the group and then proceed to the Edit Group form for the group you have just added. For information on editing groups, see "Editing Group Attributes," on page 99.

To Create a Static Group

To create a static group entry, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **New Group** link.
3. Enter the required information and click OK.

For more information, see "The New Group Page," in the online help.

Dynamic Groups

A dynamic group has an `objectclass` of `groupOfURLs`, and has zero or more `memberURL` attributes, each of which is a LDAP URL that describes a set of objects.

iPlanet Web Server enables you to create a dynamic group when you want to group users automatically based on any attribute, or when you want to apply ACLs to specific groups which contain matching DNs. For example, you can create a group that automatically includes any DN that contains the attribute `department=marketing`. If you apply a search filter for `department=marketing`, the search returns a group including all DNs containing the attribute `department=marketing`. You can then define a dynamic group from the search results based on this filter. Subsequently, you can define an ACL for the resulting dynamic group.

This section includes the following topics:

- How iPlanet Web Server Implements Dynamic Groups
- Groups Can Be Static and Dynamic
- Dynamic Group Impact on Server Performance
- Guidelines for Creating Dynamic Groups
- To Create a Dynamic Group

How iPlanet Web Server Implements Dynamic Groups

iPlanet Web Server implements dynamic groups in the LDAP server schema as `objectclass = groupOfURLs`. A `groupOfURLS` class can have multiple `memberURL` attributes, each one consisting of an LDAP URL that enumerates a set of objects in the directory. The members of the group would be the union of these sets. For example, the following group contains just one member URL:

```
ldap:///o=mcom.com??sub?(department=marketing)
```

This example describes a set that consists of all objects below "`o=mcom.com`" whose department is "marketing."

The LDAP URL can contain a search base DN, a scope and filter, however, not a hostname and port. This means that you can only refer to objects on the same LDAP server. All scopes are supported.

The DNs are included automatically, without your having to add each individual to the group. The group changes dynamically, because iPlanet Web Server performs an LDAP server search each time a group lookup is needed for ACL verification. The user and group names used in the ACL file correspond to the `cn` attribute of the objects in the LDAP database.

Note iPlanet Web Server uses the `cn` (`commonName`) attribute as group name for ACLs.

The mapping from an ACL to an LDAP database is defined both in the `dbswitch.conf` configuration file (which associates the ACL database names with actual LDAP database URLs) and the ACL file (which defines which databases are to be used for which ACL). For example, if you want base access rights on membership in a group named "staff," the ACL code looks up an object that has an object class of `groupOf<anything>` and a CN set to "staff."

The object defines the members of the group, either by explicitly enumerating the member DNs (as is done for `groupOfUniqueNames` for static groups), or by specifying LDAP URLs (for example, `groupOfURLs`).

Groups Can Be Static and Dynamic

A group object can have both `objectclass = groupOfUniqueMembers` and `objectclass = groupOfURLs`; therefore, both "`uniqueMember`" and "`memberURL`" attributes are valid. The group's membership is the union of its static and dynamic members.

Dynamic Group Impact on Server Performance

There is a server performance impact when using dynamic groups. If you are testing group membership, and the DN is not a member of a static group, iPlanet Web Server checks all dynamic groups in the database's baseDN. iPlanet Web Server accomplishes this task by checking if each `memberURL` matches by checking its baseDN and scope against the DN of the user, and then performing a base search using the user DN as baseDN and the filter of the `memberURL`. This procedure can amount to a large number of individual searches.

Guidelines for Creating Dynamic Groups

Consider the following guidelines when using the Administration Server forms to create new dynamic groups:

- Dynamic groups can not contain other groups.
- Enter the group's LDAP URL using the following format (without `host` and `port` info, since these parameters are ignored):

  ```
  ldap:///<basedn>?<attributes>?<scope>?<(filter)>
  ```

  The required parameters are described in the following table:

Table 4.5 Dynamic Groups: Required Parameters

Parameter Name	Description
<base_dn>	The Distinguished Name (DN) of the search base, or point from which all searches are performed in the LDAP directory. This parameter is often set to the suffix or root of the directory, such as "o=mcom.com".
<attributes>	A list of the attributes to be returned by the search. To specify more than one, use commas to delimit the attributes (for example, "cn,mail,telephoneNumber"); if no attributes are specified, all attributes are returned. Note that this parameter is ignored for dynamic group membership checks.
<scope>	The scope of the search, which can be one of these values: • **base** retrieves information only about the distinguished name (<base_dn>) specified in the URL. • **one** retrieves information about entries one level below the distinguished name (<base_dn>) specified in the URL. The base entry is not included in this scope. • **sub** retrieves information about entries at all levels below the distinguished name (<base_dn>) specified in the URL. The base entry is included in this scope. This parameter is required.
<(filter)>	Search filter to apply to entries within the specified scope of the search. If you are using the Administration Server forms, you must specify this attribute. Note that the parentheses are required. This parameter is required.

Note that the <attributes>, <scope>, and <(filter)> parameters are identified by their positions in the URL. If you do not want to specify any attributes, you still need to include the question marks delimiting that field.

- You can optionally also add a description for the new group.

- If any organizational units have been defined for your directory, you can specify where you want the new group to be placed using the Add New Group To list. The default location is your directory's root point, or top-most entry.
- When you are finished entering the desired information, click Create Group to add the group and immediately return to the New Group form. Alternatively, click Create and Edit Group to add the group and then proceed to the Edit Group form for the group you have just added. For information on editing groups, see "Editing Group Attributes," on page 99.

To Create a Dynamic Group

To create a dynamic group entry within the directory, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **New Group** link.
3. Select **Dynamic Group** from the Type of Group dropdown list.
4. Enter the required information and click OK.

For more information, see "The New Group Page," in the online help.

Managing Groups

The Administration Server enables you to edit groups and manage group memberships from the Manage Group form. This section describes the following topics:

- Finding Group Entries
- Editing Group Attributes
- Adding Group Members
- Adding Groups to the Group Members List
- Removing Entries from the Group Members List
- Managing Owners
- Managing See Alsos
- Removing Groups
- Renaming Groups

Finding Group Entries

Before you can edit a group entry, you must display the entry.

To find a group entry, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Groups** link.
3. Enter the name of the group that you want to find in the **Find Group** field. You can enter any of the following values in the search field:

   - A name. Enter a full name or a partial name. All entries that equally match the search string are returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
   - An asterisk (*) to see all of the groups currently residing in your directory. You can achieve the same effect by simply leaving the field blank.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.

   As an alternative, use the pull down menus in **Find all groups whose** to narrow the results of your search.
4. In the **Look within** field, select the organizational unit under which you want to search for entries. The default is the directory's root point, or top-most entry.
5. In the **Format** field, choose either On-Screen or Printer.
6. Click **Find**. All the groups matching your search criteria are displayed.
7. In the resulting table, click the name of the entry that you want to edit.

The "Find all groups whose" Field

The Find all groups whose field allows you to build a custom search filter. Use this field to narrow down the search results that are otherwise returned by Find groups. For more information regarding how to build a custom search filter, see "Building Custom Search Queries," on page 85.

To display all of the group entries contained in the **Look Within** directory, enter either an asterisk (*) or simply leave this text field blank.

Editing Group Attributes

To edit a group entry, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Groups** link.
3. Locate the group you want to edit, and type the desired changes.

   For more information regarding how to find specific entries, refer to the concepts outlined in "Finding Group Entries," on page 98.

Note You can change the Administration Server user from root to another user on the operating system to enable multiple users (belonging to the group) to edit/manage the configuration files. However, note that while on Unix/Linux platforms, the installer can give "rw" permissions to a group for the configuration files, on Windows NT platforms, the user must belong to the "Administrators" group.

For more information about editing group attributes, see "The Manage Groups Page," in the online help.

Note It is possible that you will want to change an attribute value that is not displayed by the group edit form. In this situation, use the Directory Server `ldapmodify` command line utility, if available.

Adding Group Members

To add members to a group, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Groups** link.
3. Locate the group you want to manage as described in "Finding Group Entries," on page 98, and click the **Edit** button under Group Members.

   iPlanet Web Server displays a new form that enables you to search for entries. If you want to add user entries to the list, make sure Users is shown in the **Find** pull-down menu. If you want to add group entries to the group, make sure Group is shown.
4. In the right-most text field, enter a search string. Enter any of the following options:

   - A name. Enter a full name or a partial name. All entries whose name matches the search string is returned. If no such entries are found, all entries that contain the search string are found. If no such entries are found, any entries that sounds like the search string are found.
   - A user ID if you are searching for user entries.
   - A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number are returned.
   - An email address. any search string containing an at (@) symbol is assumed to be an email address. If an exact match cannot be found, then a search is performed to find all email addresses that begin with the search string.
   - Enter either an asterisk (*) or simply leave this text field blank to see all of the entries or groups currently residing in your directory.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.
5. Click **Find and Add** to find all the matching entries and add them to the group.

If the search returns any entries that you do not want add to the group, click the box in the Remove from list? column. You can also construct a search filter to match the entries you want removed and then click **Find and Remove**.

6. When the list of group members is complete, click **Save Changes**. The currently displayed entries are now members of the group.

For more information about adding groups members, see "The Edit Members Page," in the online help.

Adding Groups to the Group Members List

You can add groups (instead of individual members) to the group's members list. Doing so causes any users belonging to the included group to become a member of the receiving group. For example, if Neil Armstrong is a member of the Engineering Managers group, and you make the Engineering Managers group a member of the Engineering Personnel group, then Neil Armstrong is also a member of the Engineering Personnel group.

To add a group to the members list of another group, add the group as if it were a user entry. For more information, see "Adding Group Members," on page 100.

Removing Entries from the Group Members List

To delete an entry from the group members list, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Groups** link, locate the group you want to manage as described in "Finding Group Entries," on page 98, and click the **Edit** button under Group Members.
3. For each member that you want to remove from the list, click the corresponding box under the Remove from list? column.

Alternatively, you can construct a filter to find the entries you want to remove and click the **Find and Remove** button. For more information on creating a search filter, see "Adding Group Members," on page 100.

4. Click **Save Changes**. The entry(s) are deleted from the group members list.

Managing Owners

You manage a group's owners list the same way as you manage the group members list. The following table identifies which section to read for more information:

Table 4.6 Additional Information

Task You Want to Complete	Read Section
Add owners to the group	"Adding Group Members," on page 100.
Add groups to the owners list	"Adding Groups to the Group Members List," on page 101.
Remove entries from the owners list	"Removing Entries from the Group Members List," on page 101.

Managing See Alsos

"See alsos" are references to other directory entries that may be relevant to the current group. They allow users to easily find entries for people and other groups that are related to the current group.

You manage see alsos the same way as you manage the group members list. The following table shows you which section to read for more information:

Table 4.7 Additional Information

Task You Want to Complete	Read Section
Add users to see alsos	"Adding Group Members," on page 100.

Table 4.7 Additional Information

Task You Want to Complete	Read Section
Add groups to see alsos	"Adding Groups to the Group Members List," on page 101.
Remove entries from see alsos	"Removing Entries from the Group Members List," on page 101.

Removing Groups

To delete a group, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Groups** link, locate the group you want to manage as described in "Finding Group Entries," on page 98, and click **Delete Group**.

Note The Administration Server does not remove the individual members of the group(s) you remove; only the group entry is removed.

Renaming Groups

To rename a group, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Groups** link and locate the group you want to manage as described in "Finding Group Entries," on page 98.
3. Click the **Rename Group** button and type the new group name in the resulting dialog box.

When you rename a group entry, you only change the group's name; you cannot use the Rename Group feature to move the entry from one organizational unit to another. For example, a business might have the following organizations:

- organizational units for Marketing and Product Management

- a group named Online Sales under the Marketing organizational unit

In this example, you can rename the group from Online Sales to Internet Investments, but you cannot rename the entry such that Online Sales under the Marketing organizational unit becomes Online Sales under the Product Management organizational unit.

Creating Organizational Units

An organizational unit can include a number of groups, and it usually represents a division, department, or other discrete business group. A DN can exist in more than one organizational unit.

To create an organizational unit, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **New Organizational Unit** link and enter the required information.

For more information, see "The New Organizational Unit Page," in the online help.

The following notes may be of interest to the directory administrator:

- New organizational units are created using the `organizationalUnit` object class.
- The distinguished name for new organizational units is of the form:

```
ou=new organization, ou=parent organization,
...,o=base organization, c=country
```

For example, if you create a new organization called Accounting within the organizational unit West Coast, and your Base DN is o=Ace Industry, `c=`US, then the new organization unit's DN is:

```
ou=Accounting, ou=West Coast, o=Ace Industry, c=US
```

Managing Organizational Units

You edit and manage organizational units from the Organizational Unit Edit form. This section describes the following tasks:

- Finding Organizational Units
- Editing Organizational Unit Attributes
- Renaming Organizational Units
- Deleting Organizational Units

Finding Organizational Units

To find organizational units, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Organizational Units** link.
3. Type the name of the unit you want to find in the Find organizational unit field. You can enter any of the following in the search field:

   - A name. Enter a full name or a partial name. All entries that equally match the search string will be returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
   - An asterisk (*) to see all of the groups currently residing in your directory. You can achieve this same result by simply leaving the field blank.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.

   As an alternative, use the pull down menus in the Find all units whose field to narrow the results of your search.
4. In the **Look within** field, select the organizational unit under which you want to search for entries. The default is the root point of the directory.
5. In the **Format** field, choose either On-Screen or Printer.

6. Click **Find**. All the organizational units matching your search criteria are displayed.

7. In the resulting table, click the name of the organizational unit that you want to find.

The "Find all units whose" Field

The Find all units whose field allows you to build a custom search filter. Use this field to narrow down the search results that are otherwise returned by Find organizational unit. For more information regarding how to build a custom search filter, see "Building Custom Search Queries," on page 85.

To display all of the group entries contained in the Look Within directory, enter either an asterisk (*) or simply leave this text field blank.

Editing Organizational Unit Attributes

To change a organizational unit entry, access the Administration Server and perform the following steps:

1. Locate the organizational unit you want to edit as described in "Finding Organizational Units," on page 105.

2. The organizational unit edit form is displayed. Change the displayed fields as desired and click **Save Changes**. The changes are made immediately.

Note It is possible that you will want to change an attribute value that is not displayed by the organizational unit edit form. In this situation, use the Directory Server `ldapmodify` command line utility, if available.

Renaming Organizational Units

To rename an organizational unit entry, access the Administration Server and perform the following steps:

1. Make sure no other entries exist in the directory under the organizational unit that you want to rename.
2. Locate the organizational unit you want to edit as described in "Finding Organizational Units," on page 105.
3. Click the **Rename** button.
4. Enter the new organizational unit name in the resulting dialog box.

Note When you rename an organizational unit entry, you can only change the organizational unit's name; you cannot use the rename feature to move the entry from one organizational unit to another. For more information, see "Renaming Organizational Units," on page 107.

Deleting Organizational Units

To delete an organizational unit entry, access the Administration Server and perform the following steps:

1. Make sure no other entries exist in the directory under the organizational unit that you want to rename.
2. Locate the organizational unit you want to delete as described in "Finding Organizational Units," on page 105.
3. Click the **Delete** button.
4. Click OK in the resulting confirmation box. The organizational unit is immediately deleted.

Managing a Preferred Language List

iPlanet Web Server enables you to display and maintain the list of preferred languages.

To manage the preferred language list, perform the following steps:

1. Access the Administration Server and choose the **Users & Groups** tab.
2. Click the **Manage Preferred Language List** link.
3. In the Display Language Selection List field, click Yes or No to specify whether iPlanet Web Server displays the Language Selection List.
4. In the **Languages in the Selection List** field, click the **Add to List** checkbox to add each language you want specified as part of the Preferred Language List.
5. Click the default value for the language you want to specify as the default language in the Preferred Language List.
6. Click **Save Changes**.

Chapter

5

Working with Server Security

This chapter describes how to activate the Secure Sockets Layer (SSL) protocol and other features designed to safeguard your data, deny intruders access, and designate who has access to the server. iPlanet Web Server incorporates the security architecture of all Netscape/iPlanet servers: it's built on industry standards and public protocols for maximum interoperability and consistency.

Before reading this chapter you should be familiar with the basic concepts of public-key cryptography. These concepts include encryption and decryption; public and private keys; digital certificates; and the SSL protocol. For more information, see *Managing Servers with Netscape Console.*

This chapter includes the following sections:

- About iPlanet Web Server Security
- Creating a New Server Instance
- Creating a Certificate Trust Database
- Requesting a Certificate
- Installing and Managing Certificates and Certificate Lists
- Using Secure Sockets Layer (SSL)
- Using Client Certificates
- Changing the Trust Database/Key Pair File Password
- Migrating Enterprise Server 3.x Certificates
- Additional Server Security Considerations

About iPlanet Web Server Security

iPlanet Web Server security is based on a number of interrelated and inter-dependent components, all of which work together to ensure that only authorized individuals can gain access to the server, that passwords or identities are not compromised, and that user identities can be trusted.

This section provides an overview for the following iPlanet Web Server security components:

- Encryption
- Certificates
- Configuring iPlanet Web Server for SSL

Encryption

Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. Decryption is the process of transforming encrypted information so that it is intelligible again. A cryptographic algorithm, also called a **cipher**, is a mathematical function used for encryption or decryption. iPlanet Web Server 4.1 includes support for various ciphers.types of ciphers.

The encryption process alone isn't enough to secure your server's confidential information. Once the information has been encrypted, and possibly transmitted to another server, a number called a **key** must be used with the encrypting cipher to produce the actual encrypted result, or to decrypt previously encrypted information. The encryption process uses two keys to achieve this result: a public key and a private key. The public key is published as part of a certificate; only the associated private key is safeguarded. (For more information about keys and certificates, see *Managing Servers with Netscape Console*.) Consequently, information encrypted with a public key can be decrypted only with the associated private key.

SSL Protocol

All Netscape/iPlanet 4.x servers support the **SSL protocol** for encrypted communication and PKCS#11 APIs for communication with software or hardware modules that perform cryptographic operations. You need to configure the Administration Server for SSL if you want to enable encryption and other cryptographic operations.

The SSL protocol supports the use of a variety of ciphers, for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. Clients and servers may support different cipher suites, or sets of ciphers, depending on factors such as the version of SSL they support, company policies regarding acceptable encryption strength, and government restrictions on export of SSL-enabled software. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suites they will use to authenticate each other, to transmit certificates, and to establish session keys.

For more information regarding how to enable SSL for iPlanet Web Server, see "Configuring Secure Sockets Layer (SSL)," on page 70 in Chapter 3, "Setting Administration Preferences."

FORTEZZA Encryption

FORTEZZA is an encryption system used by U.S. government agencies to manage sensitive but unclassified information. Use the Administration Server to configure your server to work with FORTEZZA. For information on installing the FORTEZZA hardware, see the documentation that came with your card reader.

FORTEZZA encryption support allows the web server to perform the following encryption tasks:

- use FORTEZZA ciphers for SSL connections
- use FORTEZZA card readers to store certificates and keys
- import and use FORTEZZA Compromised Key Lists (CKLs) and Certificate Revocation Lists (CRLs)
- serve files pre-encrypted with FORTEZZA ciphers

Note iPlanet Web Server, Enterprise Edition 4.1 includes FORTEZZA support for Windows NT, Solaris, and HPUX platforms.

The FORTEZZA cipher standard is a hardware smartcard standard for secure storage of private keys. The FORTEZZA operations are supported through an external PKCS#11 library. The library is added to the web server security modules database. The library handles all interfaces with external hardware (card readers) and all encryption functions. To iPlanet Web Server, Enterprise Edition, FORTEZZA support looks almost identical to any other PKCS#11 library.

Once added the security modules database (`secmod.db`), the server treats the FORTEZZA modules as any other PKCS#11 module. The FORTEZZA module is flagged as the default provider of SSL services for FORTEZZA ciphers. Then, any FORTEZZA request handled by the server is handed off to the library (via calls to NSS libraries; nothing in the actual web server code actually invokes functions in the FORTEZZA library).

The run time layer, then, is just the server and the library (no different from any PKCS#11 module).

The pre-encrypted file support runs as a web server plugin. A request for a file with a given extension (`.enc`) is routed to the plugin which invokes a function in the NSS library to send the encrypted file as a stream back to the client (no actual calls to the FORTEZZA library need to be made). The client then decrypts the data on the other side (presuming the client has a certificate with the public key corresponding to the private key that encrypted the file).

The whole configuration is managed through the Administration Server (or corresponding entries in the `magnus.conf` or `obj.conf` configuration files). The user interface enables users to select which certificate to use at run time, to collect multiple passwords (so that the server can log in to the FORTEZZA card as well as the default key database), and to allow the user to add Compromised Key Lists (CKLs)/Certificate Revocation Lists (CRLs).

When a CKL is added or updated, the certificate database is updated to make compromised keys known to the server. The NSS library validates FORTEZZA client requests against the compromised keys for each request to make sure that the client key is not a key known to be compromised.

The FORTEZZA module interoperates with the following standards and modules:

- FORTEZZA encryption standards. The FORTEZZA module allows the web server to use FORTEZZA encryption, authentication, and key validation.
- Secure Sockets Layer (SSL). FORTEZZA connections use SSL, version 3.
- The NSS libraries. Currently using NSS 2.72. The web server calls NSS functions which in turn, call the FORTEZZA library.
- The PKCS#11 web server infrastructure. FORTEZZA modules are added and configured using existing methods for managing PKCS#11 modules.
- NSAPI. The FORTEZZA module uses an NSAPI plugin to handle pre-encrypted files.
- `Modutil`: a utility for updating `secmod.db`, for adding and deleting PKCS#11 modules.

For more information regarding FORTEZZA encryption, see *Managing Servers with Netscape Console*.

FIPS-140 Compliance

You can configure iPlanet Web Server to be Federal Information Processing Standards (FIPS)-140 compliant. To make your server FIPS-140 compliant, you need to turn on the following two ciphers in your encryption preferences:

- (FIPS) DES with 56 bit encryption and SHA-1 message authentication
- (FIPS) Triple DES with 168 bit encryption and SHA-1 message authentication

You can set encryption preferences in the Administration Server by clicking the Preferences tab and the Encryption Preferences link. You can also set the encryption preferences for an instance of the iPlanet Web Server in the Server Manager by clicking the Preferences tab and the Encryption Preferences link. For more information, see “The Encryption Preferences Page,” in the online help.

Certificates

Over the Internet and many extranets and intranets, identification can take place with the aid of a **certificate**. A certificate consists of digital data that specifies the name of an individual, company, or other entity and certifies that a public key, which is also included in the certificate, belongs to that entity.

A certificate is issued and digitally signed by a **Certificate Authority**, or **CA**. A CA can be a company that sells certificates over the Internet, or it can be a department responsible for issuing certificates for your company's intranet or extranet. You decide which CAs you trust enough to serve as verifiers of other people's identities.

In addition to a public key and the name of the entity identified by the certificate, a certificate also includes additional information, such as an expiration date, the name of the CA that issued the certificate, and the "digital signature" of the issuing CA. For more information regarding the content and format of a certificate, see *Managing Servers with Netscape Console*.

Client and Server Authentication

Authentication is the process of confirming an identity. In the context of network interactions, authentication involves the confident identification of one party by another party. Certificates are one way of supporting authentication. **Client authentication** refers to the confident identification of a client by a server (that is, identification of the person assumed to be using the client software). **Server authentication** refers to the confident identification of a server by a client (that is, identification of the organization assumed to be responsible for the server at a particular network address).

Both clients and servers can have certificates. Also, clients can have multiple certificates, much like a person might have several different pieces of identification. For example, if you participate in newsgroup discussions with a Netscape Collabra Server called news.mozilla.com, you might find it possesses a certificate issued from a company named CertSafe, assuring you that this site is the one true news.mozilla.com. If you trust CertSafe's judgment, then you can trust that news.mozilla.com is the site it claims to be.

Conversely, you might be in charge of a company's internal Human Resources server. You could use your server's access-control features in conjunction with client authentication to allow only Human Resources employees access to certain directories. For more information on access control, see "What Is Access Control?," in Chapter 14, "Controlling Access to Your Server."

How iPlanet Web Server Uses Certificates to Authenticate Users

Netscape/iPlanet servers support using client certificates to authenticate a user. There are two basic ways the server can use a client certificate:

- The server matches the CA in the client certificate with a trusted CA listed in the Administration Server. This simply ensures that the client has a valid certificate from a CA the server trusts. (If the client is Netscape Navigator or Netscape Communicator and the certificate is expired, the client warns the user before sending the out-of-date certificate. Most Netscape/iPlanet servers will log an error, reject the certificate, and return a message to the client.)

- The server additionally gathers information from the client certificate and matches it with a user entry in an LDAP directory. This ensures that the client has a valid certificate and an entry in the LDAP directory. It can also ensure that the client certificate matches the one in the LDAP directory.

Note A Netscape/iPlanet server must have SSL turned on to use client certificates, and the Administration Server must trust the CA that issued the certificate to the client. For information on trusting CAs, see "Managing Certificates," on page 124.

You can configure the web server so that it refuses any client that 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. For details on configuring trusted CAs, see *Managing Servers with Netscape Console*.

Configuring iPlanet Web Server for SSL

This section explains how to get client certificate authentication working with iPlanet Web Server. When you have finished following the procedures outlined in this chapter, you will have a web server that requires a user to present a valid client SSL certificate in order to access restricted areas on the server. The certificate that the user presents must match the certificate that was published to the LDAP directory when it was issued.

This chapter focuses on setting up, installing, and managing the security components necessary to secure your iPlanet Web Server. To activate the SSL protocol for your iPlanet Web Server, you need to perform the various procedures described in the following sections:

- Creating a New Server Instance
- Creating a Certificate Trust Database
- Requesting a Certificate
- Installing and Managing Certificates and Certificate Lists
- Using Secure Sockets Layer (SSL)

Creating a New Server Instance

To use SSL with iPlanet Web Server, you must either have an existing instance of iPlanet Web Server 4.x that you want to be an SSL server or create a new instance to be an SSL server. If you have an existing instance of iPlanet Web Server that you want to simply convert to be an SSL server, you can skip this section. Otherwise, follow the steps described in this section to create a new instance of iPlanet Web Server, and then perform the remaining procedures outlined in this chapter to configure the new instance for SSL and client authentication.

To add another server instance, perform the following steps:

1. Access the Administration Server and choose the **Servers** tab.
2. Click the **Add Server** link.
3. Enter the desired information for the specified fields.

4. Click the radio button that corresponds to how you want the server to resolve IP addresses.

5. Click OK.

For more information, see "The Add Server Page," in the online help.

Creating a Certificate Trust Database

A certificate database is a key-pair and certificate database installed on the local host. When you use an internal token, the certificate database is the database into which you install the key and certificate. In iPlanet Web Server 4.x, each server instance (including the Administration Server) has its own certificate/key pair which is referred to as a **trust database**.

A **key-pair** file contains both the public and private keys used for SSL encryption. You use the key-pair file when you request and install a certificate. The key-pair file is stored encrypted in the following directory:

server_root/alias/<serverid-hostname>-key3.db.

When you create the key, you specify a password that you later use when you request the certificate and when you start a server that is using encrypted communications.

To create the certificate trust database, perform the following steps:

1. Access the Administration Server and choose the **Security** tab.

2. Type the password in **Database Password**.

3. Re-type the password in **Password** (again).

4. Click OK.

If no database exists, iPlanet Web Server creates the proper key and certificate database files and stores them in the `alias/` directory (otherwise, iPlanet Web Server displays an error message).

For more information, see "The Create a Trust Database Page," in the online help.

Requesting a Certificate

After you generate a trust database, you must create a PKCS #10 certificate request and submit it to a Certificate Authority to obtain your server SSL certificate. To enable SSL for a particular server instance, you must obtain a server SSL certificate for the server, then configure the server to require client authentication and optionally to check users' client certificates against certificate information that a CA has published to the LDAP directory.

If your company has its own internal CA for issuing certificates, you should request your certificate from them. If you plan to purchase your certificate from a commercial CA, choose a CA and ask for the specific format of the information they require. (For more information on what some CAs require, see "Required CA Information.")

Note Not everyone who requests a certificate from a commercial CA is given one. Many CAs require you to prove your identity before issuing you a certificate. Also, it can take anywhere from a day to two months or more to approve a certificate. You are responsible for promptly providing all the necessary information to the CA.

To request a certificate, make sure you know what information your CA requires, and then perform the following steps:

1. Access the Administration Server and choose the **Security** tab.
2. Click the **Request Certificate** link.
3. In the form that iPlanet Web Server displays, specify if this is a new certificate or a certificate renewal. Many certificates expire after a set period of time, such as six months or a year. Some CAs will automatically send you a renewal.

4. Perform the following steps to specify how you want to submit the request for the certificate:

    1. If the CA expects to receive the request in an email message, check CA Email and enter the email address of the CA. For a list of CAs, click List of available certificate authorities.

    2. If you are requesting the certificate from an internal CA that is using Netscape Certificate Server, click **CA URL** and enter the URL for the Certificate Server. This URL should point to the certificate server's program that handles certificate requests. A sample URL might be: `https://CA.mozilla.com:444/cms`.

5. From the drop-down list, select the cryptographic module for the key-pair file you want to use when requesting the certificate.

    For information about installing additional cryptographic modules, see "Adding a PKCS#11Module," on page 128, or see "The Install a New PKCS#11 Module Page," in the online help.

6. Type the password for your key-pair file. This is the same password you specified when you created the trust database in "Creating a Certificate Trust Database." The server uses the password to get your private key and encrypt a message to the CA. The server then sends both your *public key* and the encrypted message to the CA. The CA uses the public key to decrypt your message.

7. Type your identification information. The format of this information varies by CA. For a general description of these fields, see "Required CA Information." Note that most of this information usually isn't required for a certificate renewal.

8. Double-check your work to ensure accuracy. The more accurate the information, the faster your certificate is likely to be approved.

9. Click OK once you've checked that the information is correct. If your request is going to a certificate server, you'll be prompted to verify the form information before the request is submitted. You should re-read the information and then click OK to submit the request to the certificate server.

For more information, see "The Request a Server Certificate Page," in the online help.

The server generates a certificate request that contains your information. The request has a digital signature created with your private key. The CA uses a digital signature to verify that the request wasn't tampered with during routing from your server machine to the CA. In the rare event that the request is tampered with, the CA will usually contact you by phone.

If you chose to email the request, the server composes an email message containing the request and sends the message to the CA. Typically, the certificate is sent to you via email. If instead you specified a URL to a certificate server, your server uses the URL to submit the request to the Certificate Server. You might get a response via email or other means depending on the CA.

If the CA agrees to issue you a certificate, the CA will notify you. (In most cases, the CA will send your certificate via email. If your organization is using a certificate server, you may be able to search for the certificate by using the certificate server's forms.)

Once you receive the certificate, you can install it. In the meantime, you can still use your server without SSL.

Required CA Information

Whether you are requesting a server certificate from a commercial CA or an internal CA, you need to provide the following information:

- **Common Name** must be the fully qualified hostname used in DNS lookups (for example, *www.iplanet.com*). This is the hostname in the URL that a browser uses to connect to your site. It's important that these two names are the same, otherwise a client is notified that the certificate name doesn't match the site name, which will make people doubt the authenticity of your certificate. However, some CAs might require different information, so it's important to contact them. Note that you can not use wildcards in a common name.

- **Email Address** is your business email address. This is used for correspondence between you and the CA.

- **Organization** is the official, legal name of your company, educational institution, partnership, and so on. Most CAs require that you verify this information with legal documents (such as a copy of a business license).

- **Organizational Unit** is an optional field that describes an organization within your company. This can also be used to note a less formal company name (without the *Inc.*, *Corp.*, and so on).
- **Locality** is an optional field that usually describes the city, principality, or country for the organization.
- **State or Province** is usually required, but can be optional for some CAs. Note that most CAs won't accept abbreviations, but check with them to be sure.
- **Country** is a required, two-character abbreviation of your country name (in ISO format). The country code for the United States is US.

All this information is combined as a series of attribute-value pairs called the distinguished name (DN), which uniquely identifies the subject of the certificate.

If you are purchasing your certificate from a commercial CA, you must contact the CA to find out what additional information they require before they issue a certificate. Most CAs require that you prove your identity. For example, they want to verify your company name and who is authorized by the company to administer the server, and they might ask whether you have the legal right to use the information you provide.

Some commercial CAs offer certificates that indicate a greater level of detail and veracity to vendors or individuals who provide greater proof of their identity. For example, you might be able to purchase a certificate stating that the CA has not only verified that you are the rightful administrator of the www.mozilla.com computer, but that you really are a company that has been in business for ten years and have no outstanding customer litigation against you. Generally, these certificates cost more than standard ones.

Installing and Managing Certificates and Certificate Lists

This section includes the following topics:

- Installing Certificates
- Managing Certificates
- Managing Certificate Lists

Installing Certificates

There are three types of certificates that you can install:

- Your own server's certificate to present to clients.
- A CA's own certificate for use in a certificate chain.

Each of these certificates is installed through the process described here.

When you receive a certificate from the CA, it will be encrypted with your public key so that only you can decrypt it. The server will use the key-pair file password you specify to decrypt the certificate when you install it. You can either save the email somewhere accessible to the server, or copy the text of the email and be ready to paste the text into the Install Certificate form, as described here.

A **certificate chain** is a hierarchical series of certificates signed by successive certificate authorities. A CA certificate identifies a certificate authority (CA) and is used to sign certificates issued by that authority. A CA certificate can in turn be signed by the CA certificate of a parent CA and so on up to a root CA.

Note CAs' certificates for use in a certificate chain are installed using the same process as installing your own certificate. If your CA doesn't automatically send you their certificate, you should request it. However, many CAs include their certificate in the same email that contains your certificate. In this case, your server installs both certificates at the same time when you install your certificate. For more information on certificate chaining, see "Appendix D Introduction to Public-Key Cryptography," in *Managing Servers with Netscape Console.*

To install a certificate and associate it with an alias, perform the following steps:

1. Access the Administration Server and choose the **Security** tab.
2. Click the **Install Certificate** link.
3. Check the type of certificate you are installing:
   - **This Server** is for a single certificate associated only with your server.
   - **Server Certificate Chain** is for a CA's certificate to include in a certificate chain.
   - **Trusted Certificate Authority (CA)** is for a certificate of a CA that you want to accept as a trusted CA for client authentication.
4. If the certificate is for a chain, name the certificate. iPlanet Web Server displays this name in the Manage Certificates list. The name should be descriptive and can include spaces. For example, "United States Postal Service CA" is the name of the CA, and "VeriSign Class 2 Primary CA" describes both the CA and the type of certificate. If the certificate is for "this server," the Administration Server uses the name Server-Cert.
5. Either type the full pathname to the saved email or paste the email text in the field called **Message text (with headers)**. If you copy and paste the text, be sure to include the headers "Begin Certificate" and "End Certificate"—including the beginning and ending hyphens. Make sure you check the corresponding radio button for either the file or the text.
6. Click **OK**.
7. Click **Add**.

   The certificate is stored in the server's certificate database. The filename will be `<alias>-cert.db`. For example:

   `https-<serverid>-<hostname>-cert7.db`

   For more information, see "The Install a Server Certificate Page," in the online help.

If you have just installed your own certificate, you can now activate SSL for your server. To activate SSL, see "Activating SSL," on page 127.

Managing Certificates

You can view, delete, or edit the trust settings of all the certificates installed on your server. This includes your own certificate and certificates from CAs.

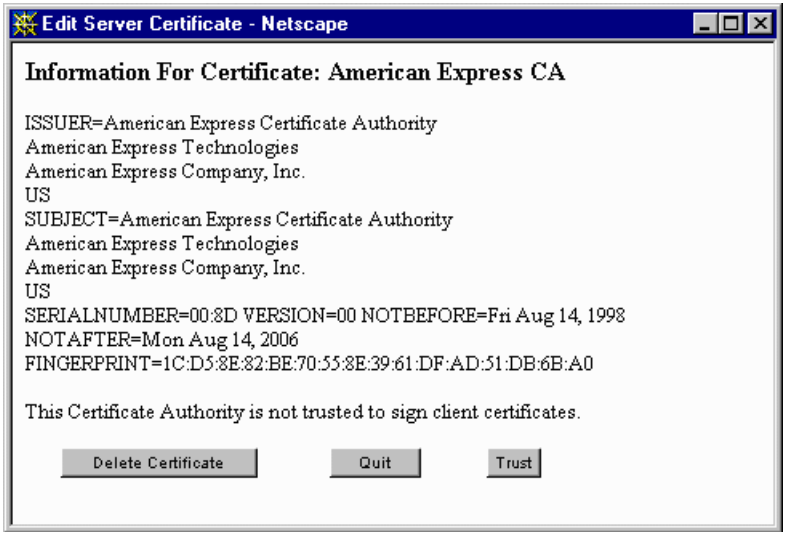
To manage this list of certificates, perform the following steps:

1. Access the Administration Server and choose the **Security** tab.
2. Click the **Manage Certificates** link.

   iPlanet Web Server displays the Manage Server Certificates page.
3. Select a certificate file alias, and then click OK.

   All of the installed certificates associated with the alias appear with their type and expiration date. The link text is the name given to the certificate when it was installed. The Administration Server comes with several default certificates, which are listed above the certificates you installed. all certificates are stored in the directory *server_root*/alias.
4. To view more information about a certificate, click the link for the certificate. A window appears, containing information about that certificate. Figure 5.1 shows a sample.

Figure 5.1 Certificate information includes the owner and who issued it.



5. To trust the CA, click **Trust**. If the CA is already trusted, you can click **Do Not Trust**. By default, all CAs are not trusted.

   To delete the certificate, click the **Delete Certificate** button.

   To close the window, click the **Quit** button.

For more information, see "The Manage Server Certificates Page (Administration Server)," in the online help.

Note that trust settings refer specifically to whether a certificate is trusted as a signer of client certificates (the user does not, for example, have to trust a CA after the CA issues a server certificate).

Managing Certificate Lists

The purpose of certificate revocation lists (CRLs) and compromised key lists (CKLs) is to make known any certificates and keys that either client or server users should no longer trust. If data in a certificate changes (for example, a user changes offices or leaves the organization) before the certificate expires, the certificate is revoked and its data appears in a CRL. If a key is tampered with or otherwise compromised, the key and its data appear in a CKL. Both CRLs and CKLs are produced and periodically updated by a CA.

This section includes the following topics:

- Obtaining a CRL or CKL
- Adding a CRL or CKL to the Trust Database
- Managing CRLs

Obtaining a CRL or CKL

To obtain a CRL or CKL from a Certificate Authority (CA), perform the following steps:

1. Use a browser to go to the CA's web site. Contact your CA administrator for the exact URL to use.

2. Follow the CA's instructions for downloading the CRL or CKL to a local directory.

Once you've saved the CRL file or CKL file to a local directory, you can add information from it to the Trust Database.

Adding a CRL or CKL to the Trust Database

To add CRL or CKL to the trust database, perform the following steps:

1. Access the Administration Server and choose the **Security** tab.
2. Click the **Install CRL/CKLs** link.

   iPlanet Web Server displays the Install a Certificate Revocation List page.
3. Click the **File contains** radio button for either **Certificate Revocation List** or **Compromised Key List**.
4. In **The CRL/CKL is in this file** field, type the full path name to the associated file.
5. Click OK. If the list already exists in the database, the list you specify here will replace the existing list.

Managing CRLs

To manage CRLs, perform the following steps:

1. Access the Administration Server and choose the **Security** tab.
2. Click the **Manage CRLs** link.

   iPlanet Web Server displays the Manage Certificate Revocation Lists page. All installed CRLs are listed along with their expiration dates.
3. Click on the desired CRL for more information and options.
4. Click a CRL to select it.
5. To add the CRL to the trust database, click **Add**.

To delete CRL from the trust database, click **View**. In the Certificate window, click **Delete**.

Using Secure Sockets Layer (SSL)

After you have generated a key-pair file and installed your certificate, you can activate SSL for your Administration Server or any other iPlanet Web Server.

This section includes the following topics:

- Activating SSL
- Specifying Ciphers
- Setting Security (SSL) Preferences
- Adding a PKCS#11Module
- Using SSL Configuration File Directives

Activating SSL

To activate SSL for iPlanet Web Server, perform the steps described in "Activating SSL," on page 71 in Chapter 3, "Setting Administration Preferences."

URLs to an SSL-enabled iPlanet Web Administration 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]]:[port#]>
```

For example, `https://admin.mozilla.com:443`. If you use the default secure http port number (443), you don't have to use the port number in the URL.

Specifying 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.

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. Improvements were made to the protocol after version 2 that improve security and performance; you should not use SSL 2 unless you have a real need to service clients that are not capable of using SSL 3. Client certificates are not guaranteed to work with SSL 2 ciphers. 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.

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

For more information regarding specific ciphers, see *Managing Servers with Netscape Console*.

Setting Security (SSL) Preferences

You can set preferences for using SSL encryption on any server. To set the SSL preferences for iPlanet Web Server, perform the steps described in "Setting Encryption Preferences," on page 71 in Chapter 3, "Setting Administration Preferences."

Adding a PKCS#11 Module

iPlanet Web Server supports Public Key Cryptography Standard (PKCS) #11, which defines the interface used for communication between SSL and PKCS#11 modules. The PKCS#11 modules are used for standards-based connectivity to SSL hardware accelerators. You can import PKCS#11 modules in the form of `.jar` files or object files.

Guidelines for Installing a PKCS#11 Module

Even though you install an external PKCS#11 module, you still must create a Trust Database using the Internal (software) module. The PKCS#11 and SSL code relies on the default certificate and key databases.

If you do not create a Trust Database (using the Security tab "Create Database" link), one will be created for you when you request or install a certificate for an external PKCS#11 module. However, when a module is created for you, it has no password and cannot be accessed. This means that your external module will continue to work, but that you will not be able to create and install server certificates using the internal PKCS#11 module in the future.

For reference: If you allow a default database to be created without a password and later discover you want to use the internal PKCS#11 module, you can simply delete the existing database files:

```
$SERVER_ROOT/alias/https-$SERVERID-$HOSTNAME-key3.db

$SERVER_ROOT/alias/https-$SERVERID-$HOSTNAME-cert7.db
```

For example, for the server named *secure.example.com* installed in

```
/usr/local/netscape
```

the files would be:

```
/usr/local/netscape/alias/https-secure.example.com-
secure-key3.db

/usr/local/netscape/alias/https-secure.example.com-
secure-cert7.db
```

After deleting the existing databases, you can re-create them using the Security tab **Create Database** link.

If you install a certificate for your server into an external PKCS#11 module (for example, a hardware accelerator), the server will not be able to start using that certificate until you manually edit `magnus.conf`.

The server always tries to start with the certificate named "Server-Cert." However, certificates in external PKCS#11 modules include one of the module's token names in their identifier. For example, a sever certificate installed on an external smartcard reader called "smartcard0" would be named "smartcard0:Server-Cert."

To tell the server to start with that server certificate instead, you must edit `magnus.conf` and add the following line anywhere in the file:

```
CERTDefaultNickname $TOKENNAME:Server-Cert
```

To find out what value to use for `$TOKENNAME`, go to the server's Security tab and select the Manage Certificates link. When you log in to the external module where Server-Cert is stored, its certificates are displayed in the list in the `$TOKENNAME:$NICKNAME` form.

To Import a PKCS#11 Module

To import a PKCS#11 module, perform the following steps:

1. Access the Administration Server and choose the **Security** tab.
2. Click the **Add PKCS#11 Module** link.
3. Type the path for the .jar file in **Path to Jar File**.
4. Click OK.

For information on using PKCS#11, see "The Install a New PKCS#11 Module Page," in the online help.

Adding a FORTEZZA PKCS#11 Module

The library is the only integration per se done on the server side. The server gets use of the library for "free", as it appears like any other PKCS#11 library to the server. The user may be required to do further integration offline to ensure that the FORTEZZA library integrates with drivers for the FORTEZZA card reader.

Once the library is installed, the you need to acquire FORTEZZA credentials (a card with certificates) offline.

Note The library must be installed with `-ciphers FORTEZZA` so that the NSS library recognizes it as the default service provider for FORTEZZA encryption.

The iPlanet Web Server is used to enable FORTEZZA ciphers, select a certificate from a FORTEZZA card for the server to use, and install a Compromised Key List (CKL). A CKL is the current list of revoked key material. The CKL needs to be loaded into the certificate database on both the client and the 4.x servers in order to use the FORTEZZA ciphers.

It is possible to run a server that uses a FORTEZZA certificate and an RSA certificate. There is no explicit integration required. The user will be able to select more than one certificate to be used as the server certificate. At connection time, the NSS libraries will handle selecting the appropriate certificate for the client connection during the SSL handshake.

If pre-encrypted file support is to be used, `obj.conf` needs to be modified to load the plugin.

Note If you have the FORTEZZA ciphers enabled on both your client and the server, the common cipher suite of communication should be FORTEZZA. This can be checked using the Page Info on your client. Otherwise, there should not be any difference in the behavior of the server.

For more information regarding FORTEZZA encryption, see *Managing Servers with Netscape Console*.

Using SSL Configuration File Directives

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

Security

The `Security` tells the server whether encryption (Secure Sockets Layer version 2 or version 3 or both) 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.

If `Security` is set to on, and both SSL2 and SSL3 are enabled, then the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption. By default, security is off.

SSL2

The `SSL2` directive tells the server whether Secure Sockets Layer, version 2 encryption is enabled or disabled. The `Security` directive dominates the `SSL2` directive; if `SSL2` encryption is enabled but the `Security` directive is set to off, then it is as though `SSL2` were 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. By default, security is off.

SSL3

The `SSL3` directive tells the server whether Secure Sockets Layer, version 3 security is enabled or disabled. The `Security` directive dominates the `SSL3` directive; if `SSL3` security is enabled but the `Security` directive is set to off, then it is as though `SSL3` were 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. By default, security is off.

Ciphers

The `Ciphers` directive specifies the ciphers enabled for your server.

Syntax `Ciphers +rc4,+rc4export,+rc2,+rc2export,+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.

Syntax `SSL3Ciphers +fortezza,+fortezza_null_md5,+rsa_rc4_128_md5,+rsa_3des_sha,+rsa_des_sha,+rsa_rc4_40_md5,+rsa_rc2_40_md5,rsa_null_md5,+rsa_des_56_sha,+rsa_rc4_56_sha`

`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.

SSL3SessionTimeout

The SSL3SessionTimeout directive controls SSL3 session caching.

Syntax `SSL3SessionTimeout` seconds

`seconds` is the number of seconds until a cached SSL3 session becomes invalid. The default value is 86400 (24 hours). If the `SSL3SessionTimeout` directive is specified, the value of seconds is silently constrained to be between 5 and 86400 seconds.

SSLCacheEntries

Specifies the number of SSL sessions that can be cached.

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.

SSLSessionTimeout

The SSLSessionTimeout directive controls SSL2 session caching.

Syntax `SSLSessionTimeout` *`seconds`*

`seconds` is the number of seconds until a cached SSL2 session becomes invalid. The default value is 100. If the `SSLSessionTimeout` directive is specified, the value of seconds is silently constrained to be between 5 and 100 seconds.

Using Client Certificates

If you have enabled the Administration Server Preferences "Require client certificates" option, 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. For more information, see "Access Control Files," on page 341 in Chapter 14, "Controlling Access to Your Server." In addition, you can process information from client certificates. For more information, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

Mapping Client Certificates to LDAP

This section describes the process iPlanet Web Server uses to map a client certificate to an entry in an LDAP directory.

When the server gets a request from a client, it asks for the client's certificate before proceeding. Netscape clients, such as Netscape Navigator and Netscape Communicator, send the client certificate to the server (with or without prompting the end user, depending on the browser's security configuration). (Note that you also need to set up the required ACLs; for more information, see "ACL File Syntax," in Appendix B, "ACL File Syntax," on page 466).

The server then takes the CA listed in the certificate and tries to match it to a trusted CA listed in the Administration Server. If there isn't a match, some servers end the connection and some perform a different operation based on the failed match. iPlanet Web Server ends the connection. If there is a match, the server continues processing the request.

After the server checks that the certificate's CA is trusted, the server performs the following steps to map the certificate to an LDAP entry:

1. Maps the subject (user's) DN from the user's cert to a branch point in the LDAP directory.
2. Searches the LDAP directory for an entry that matches the information about the subject (end-user) of the client certificate.
3. Optionally verifies the client certificate with one in the LDAP entry that corresponds to the DN.

The server uses a certificate mapping file called `certmap.conf` to determine how to do the LDAP search. The mapping file tells the server what values to take from the client certificate (such as the end-user's name, email address, and so on). The server uses these values to search for a user entry in the LDAP directory, but first the server needs to determine where in the LDAP directory it needs to start its search. The certificate mapping file also tells the server where to start.

Once the server knows where to start its search and what it needs to search for (step 1), it performs the search in the LDAP directory (step 2). If it finds no matching entry or more than one matching entry, and the mapping is *not* set to verify the certificate, the search fails. For a complete list of the expected search result behavior, see the following LDAP Search Results table. Note that you can specify the expected behavior in the ACL; for example, you can specify that iPlanet Web Server accepts only you if the certificate match fails. For more information regarding how to set the ACL preferences, see "Access Control Files," on page 341 in Chapter 14, "Controlling Access to Your Server."

Table 5.1 LDAP Search Results

LDAP Search Result	Certificate Verification ON	Certificate Verification OFF
No entry found	Authorization fails	Authorization fails
Exactly one entry found		Authorization succeeds
More than one entry found		Authorization fails

After the server finds a matching entry and certificate in the LDAP directory, it can use that information to process the transaction. For example, some servers use certificate-to-LDAP mapping to determine access to a server.

The following section describes the `certmap.conf` file. You need to edit this file to fit the entries in your LDAP directory and to match the certificates you expect your users to have.

Using the certmap.conf File

The certificate mapping file determines how a server should look up a user entry in the LDAP directory. You edit this file and add entries to match the organization of your LDAP directory and to list the certificates you want your users to have. Specifically, the mapping file defines the following information:

- where in the LDAP tree the server should begin its search
- what certificate attributes the server should use as search criteria when searching for the entry in the LDAP directory
- whether or not the server goes through an additional verification process

The certificate mapping file is located in the following location:

```
server_root/userdb/certmap.conf
```

The file contains one or more named mappings, each applying to a different CA. A mapping has the following syntax:

```
certmap <name> <issuerDN>

<name>:<property> [<value>]
```

The first line specifies a name for the entry and the attributes that form the distinguished name found in the CA certificate. The name is arbitrary; you can define it to be whatever you want. However, `issuerDN` must *exactly* match the issuer DN of the CA who issued the client certificate. For example, the following two `issuerDN` lines differ only in the spaces separating the attributes, but the server treats these two entries as different:

```
certmap moz1 ou=Mozilla Certificate
Authority,o=Netscape,c=US
certmap moz2 ou=Mozilla Certificate Authority,
o=Netscape, c=US
```

The second and subsequent lines in the named mapping match properties with values. The `certmap.conf` file has six default properties (you can use the certificate API to customize your own properties):

- `DNComps` is a list of comma-separated attributes used to determine where in the LDAP directory the server should start searching for entries that match the user's information (that is, the owner of the client certificate). The server gathers values for these attributes from the client certificate and uses the values to form an LDAP DN, which then determines where the server starts its search in the LDAP directory. For example, if you set `DNComps` to use the `o` and `c` attributes of the DN, the server starts the search from the `o=<org>, c=<country>` entry in the LDAP directory, where `<org>` and `<country>` are replaced with values from the DN in the certificate.

  Note the following situations:

  - If there isn't a `DNComps` entry in the mapping, the server uses either the `CmapLdapAttr` setting or the entire subject DN in the client certificate (that is, the end-user's information).

  - If the `DNComps` entry is present but has no value, the server searches the entire LDAP tree for entries matching the filter.

- `FilterComps` is a list of comma-separated attributes used to create a filter by gathering information from the user's DN in the client certificate. The server uses the values for these attributes to form the search criteria used to match entries in the LDAP directory. If the server finds one or more entries in the LDAP directory that match the user's information gathered from the certificate, the search is successful and the server optionally performs a verification.

  For example, if `FilterComps` is set to use the email and userid attributes (`FilterComps=e,uid`), the server searches the directory for an entry whose values for email and uid match the end user's information gathered from the client certificate. Email addresses and userids are good filters because they are usually unique entries in the directory. The filter needs to be specific enough to match one and only one entry in the LDAP database.

For a list of the x509v3 certificate attributes, see the following table:

Table 5.2 Attributes for x509v3 Certificates

Attribute	Description
`c`	Country
`o`	Organization
`cn`	Common name
`l`	Location
`st`	State
`ou`	Organizational unit
`uid`	Unix/Linux userid
`e|mail`	Email address

Note that the attribute names for the filters need to be attribute names from the certificate, not from the LDAP directory. For example, some certificates have an `e` attribute for the user's email address; whereas LDAP calls that attribute `mail`.

- `verifycert` tells the server whether it should compare the client's certificate with the certificate found in the LDAP directory. It takes two values: on, and off. You should only use this property if your LDAP directory contains certificates. This feature is useful to ensure your end-users have a valid, unrevoked certificate.

- `CmapLdapAttr` is a name for the attribute in the LDAP directory that contains subject DNs from all certificates belonging to the user. The default for this property is `certSubjectDN`. This attribute isn't a standard LDAP attribute, so to use this property, you have to extend the LDAP schema. For more information, see *Managing Servers with Netscape Console*.

  If this property exists in the `certmap.conf` file, the server searches the entire LDAP directory for an entry whose attribute (named with this property) matches the subject's full DN (taken from the certificate). If the search doesn't find any entries, the server retries the search using the `DNComps` and `FilterComps` mappings.

  This approach to matching a certificate to an LDAP entry is useful when it's difficult to match entries using `DNComps` and `FilterComps`.

- `Library` is a property whose value is a pathname to a shared library or DLL. You only need to use this property if you create your own properties using the certificate API. For more information, see the *NSAPI Programmer's Guide for iPlanet Web*.
- `InitFn` is a property whose value is the name of an init function from a custom library. You only need to use this property if you create your own properties using the certificate API.

For more information on these properties, refer to the examples described in "Example Mappings," on page 139

Creating Custom Properties

You can use the client certificate API to create your own properties. For information on programming and using the client certificate API, see *NSAPI Programmer's Guide for iPlanet Web Server*.

Once you have a custom mapping, you reference the mapping as follows:

<*name*>:`library` <*path_to_shared_library*>
<*name*>:`InitFn` <*name_of_init_function*>

For example:

```
certmap default1 o=Netscape Communications, c=US
default1:library /usr/netscape/enterprise/userdb/
plugin.so
default1:InitFn plugin_init_fn
default1:DNComps  ou o c
default1:FilterComps l
default1:verifycert on
```

Example Mappings

The `certmap.conf` file should have at least one entry. The following examples illustrate the different ways you can use the `certmap.conf` file.

Example #1

This example represents a `certmap.conf` file with only one "default" mapping:

```
certmap default default
default:DNComps ou, o, c
default:FilterComps e, uid
default:verifycert on
```

Using this example, the server starts its search at the LDAP branch point containing the entry `ou=<orgunit>, o=<org>, c=<country>` where the text in `<>` is replaced with the values from the subject's DN in the client certificate.

The server then uses the values for email address and userid from the certificate to search for a match in the LDAP directory. When it finds an entry, the server verifies the certificate by comparing the one the client sent to the one stored in the directory.

Example #2

The following example file has two mappings: a default one and another for the US Postal Service:

```
certmap default default
default:DNComps
default:FilterComps e, uid

certmap usps ou=United States Postal Service, o=usps,
c=US
usps:DNComps ou,o,c
usps:FilterComps e
usps:verifycert on
```

When the server gets a certificate from anyone other than the US Postal Service, it uses the default mapping, which starts at the top of the LDAP tree and searches for an entry matching the client's email and userid. If the certificate is from the US Postal Service, the server starts its search at the LDAP branch containing the organizational unit and searches for matching email addresses. Also note that if the certificate is from the USPS, the server verifies the certificate; other certificates are not verified.

Warning The issuer DN (that is, the CA's information) in the certificate must be identical to the issuer DN listed in the first line of the mapping. In the previous example, a certificate from an issuer DN that is `o=United States Postal Service,c=US` won't match because there isn't a space between the o and the `c` attributes.

Example #3

The following example uses the `CmapLdapAttr` property to search the LDAP database for an attribute called `certSubjectDN` whose value exactly matches the entire subject DN taken from the client certificate.

```
certmap myco ou=My Company Inc, o=myco, c=US
myco:CmapLdapAttr certSubjectDN
myco:DNComps  o, c
myco:FilterComps mail, uid
myco:verifycert on
```

If the client certificate subject is:

```
uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

the server first searches for entries that contain the following information:

```
certSubjectDN=uid=Walt Whitman, o=LeavesOfGrass Inc,
c=US
```

If one or more matching entries are found, the server proceeds to verify the entries. If no matching entries are found, the server will use `DNComps` and `FilterComps` to search for matching entries. In this example, the server would search for `uid=Walt Whitman` in all entries under `o=LeavesOfGrass Inc, c=US`.

Note This example assumes the LDAP directory contains entries with the attribute `certSubjectDN`.

Changing the Trust Database/Key Pair File Password

It's a good practice to change your trust database/key pair file password periodically. If your Administration Server is SSL enabled, this password is required when starting the server. Changing your password periodically adds an extra level of server protection.

For a list of guidelines to consider when changing a password, see "Guidelines for Creating Hard-to-Crack Passwords," on page 144

To change your trust database/key pair file) password, perform the following steps:

1. Access the Administration Server and choose the **Security** tab.
2. Click the **Change Password** link.
3. Type the required information and click OK.

For more information, see "The Change the Key Pair File Password Page," in the online help.

Migrating Enterprise Server 3.x Certificates

If you need to migrate certificates from an Enterprise Server 3.6 to iPlanet Web Server 4.x make sure that the 4.x iPlanet Web Administration Server user has read and write permissions on the old 3.6 database files. The files are `<alias>-cert.db` and `<alias>-key.db`, located in the `<3.6_server_root>/alias` directory.

Keys and certificates are migrated as part of the migration process only if your server has security enabled. You can also migrate keys and certificates by themselves using the Security tabs in the Administration Server page and the Server Manager page.

In Enterprise Server 3.6, a certificate/key pair was referred to by an alias which could be used by multiple server instances. The administration server managed all the aliases and their constituent certificates. In iPlanet Web Server 4.x, each server instance (including the Administration Server) has its own certificate/key pair which is referred to as a trust database instead of an alias. You manage the trust database and its constituent certificates, including the server certificate and all the included Certificate Authorities, from the Server Manager's Security tab. The certificate and key database files are now named after the server instance that uses them. If multiple 3.6 server instances use the same alias, when you migrate each instance the certificate/key pair are migrated and named for the new server instance.

The migration not only migrates the server certificate, it migrates the whole trust database associated with the server instance. All the Certificate Authorities (CAs) in your 3.6 database are migrated to the 4.x database. If they duplicate the 4.x CAs, you use the 3.6 CA until it expires, then the 4.x CA. Do not attempt to delete duplicate CAs.

Additional Server Security Considerations

There are other security risks besides someone trying to break your encryption. Networks face risks 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. For example, put the server machine into a secure room, and don't allow untrusted individuals to upload programs to your server.

The following sections describe the most important things you can do to make your server more secure:

- Limit Physical Access
- Limit Administration Access
- Choose Good Passwords
- Secure Your Key-Pair File
- Limit Other Applications on the Server
- Prevent Clients from Caching SSL Files
- Limit Ports
- Know Your Server's Limits
- Consider Additional Measures for Unprotected Servers

Limit Physical Access

This simple security measure is often forgotten. Keep the server machine in a locked room that only authorized people can enter. This prevents anyone from hacking the server machine itself.

Also, protect your machine's administrative (root) password, if you have one.

Limit Administration Access

If you use remote configuration, be sure to use access control to allow administration from only a few users and computers. If you want your Administration Server to provide end-user access to the LDAP server or local directory information, consider maintaining two Administration Servers and using cluster management so that the SSL-enabled Administration Server acts as the master server and the other Administration Server is available for end-users' access. For more information regarding clusters, see "About Clusters," on page 150 in Chapter 6, "Managing Server Clusters."

You should also turn on encryption for the Administration Server. If you don't use an SSL connection for administration, then you should be cautious when performing remote server administration over an unsecure network. Anyone could intercept your administrative password and reconfigure your servers.

Choose Good Passwords

You use a number of passwords with your server—the administrative password, the private key password, database passwords, and so on. Your administrative password is the most important password of all, since anyone with that password can configure any and all servers on your computer. Most important after that is your private key password. If someone gets your private key and your private key password, they can create a fake server that appears to be yours, or intercept and change communications to and from your server.

A good password is one you'll remember but others won't guess. For example, you could remember `MCi12!mo` as "My Child is 12 months old!" A bad password is your child's name or birthdate.

Guidelines for Creating Hard-to-Crack Passwords

There are some simple guidelines that will help you create a stronger password.

It is not necessary to incorporate all of the following rules in one password, but the more of the rules you use, the better your chances of making your password hard to crack:

- Passwords should be 6-14 characters long.

- Mac passwords cannot be longer than 8 characters.
- Do not use the “illegal” characters: *, ", or spaces.
- Do not use dictionary words (any language).
- Do not make common letter substitutions (like replacing 3’s for E’s and 1’s for L’s) within dictionary words.
- Include characters from as many of these classes as possible:
  - Uppercase letters
  - Lowercase letters
  - Numbers
  - Symbols

Secure Your Key-Pair File

Make sure your key-pair file is protected. The Administration Server stores key-pair files in the directory *server_root*/alias. Consider making the files and directory readable only to Netscape/iPlanet servers installed on your computer. It’s also important to know if the file is stored on backup tapes or is otherwise available for someone to intercept. If so, you must protect your backups as completely as your server.

Limit Other Applications on the Server

Carefully consider all applications that run on the same machine as the server. It’s possible to circumvent your server’s security by exploiting holes in other programs running on your server. Disable all unnecessary programs and services. For example, the Unix `sendmail` daemon is difficult to configure securely and it can be programmed to run other possibly detrimental programs on the server machine.

Unix/Linux Carefully choose the processes started from `inittab` and `rc` scripts. Don’t run `telnet` or `rlogin` from the server machine. You also shouldn’t have `rdist` on the server machine (this can distribute files but it can also be used to update files on the server machine).

Windows NT Carefully consider which drives and directories you share with other machines. Also, consider which users have accounts or Guest privileges.

Similarly, be careful about what programs you put on your server or allow other people to install on your server. Other people's programs might have security holes. Worst of all, someone might upload a malicious program designed specifically to subvert your security. Always examine programs carefully before you allow them on your server.

Prevent 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">
```

Limit Ports

Disable any ports not used on the machine. Use routers or firewall configurations to prevent incoming connections to anything other than the absolute minimum set of ports. This means that the only way to get a shell on the machine is to physically use the server's machine, which should be in a restricted area already.

Know Your Server's Limits

The server offers secure connections between the server and the client. It can't control the security of information once the client has it, nor can it control access to the server machine itself and its directories and files.

Being aware of these limitations helps you know what situations to avoid. For example, you might acquire credit card numbers over an SSL connection, but are those numbers stored in a secure file on the server machine? What happens to those numbers after the SSL connection is terminated? You should be responsible for securing any information clients send to you through SSL.

Consider Additional Measures for Unprotected Servers

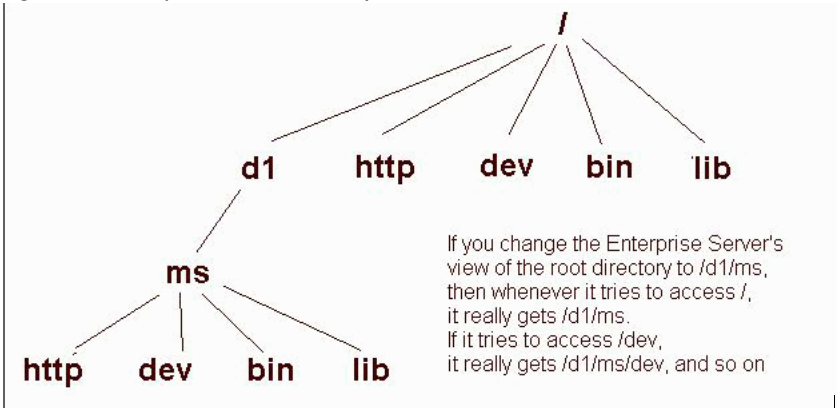
If you want to have both protected and unprotected servers, you should operate the unprotected server on a different machine from the protected one. If your resources are limited and you must run an unprotected server on the same machine as your protected server, do the following.

- 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.
- For Unix/Linux, enable the `chroot` feature for the document root directory. The unprotected server should have references to its document root redirected using `chroot`.

The purpose of `chroot` is to allow you to create a second root directory to limit the server to specific directories. You'd use this feature to safeguard an unprotected server. For example, you could say that the root directory is `/d1/ms`. Then any time the web server tries to access the root directory, it really gets `/d1/ms`. If it tries to access `/dev`, it gets `/d1/ms/dev` and so on. This allows you to run the web server on your Unix/Linux system, without giving it access to all the files under the actual root directory.

However, if you use `chroot`, you need to set up the full directory structure that iPlanet Web Server needs, under the alternative root directory, as shown in the following illustration:

Figure 5.2 Example Chroot Directory Structure



For more information regarding how to implement `chroot` in the `magnus.conf` file, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

Chapter

6

Managing Server Clusters

This chapter describes the concept of clustering iPlanet Web server s and explains how you can use them to share configurations among servers.

This chapter includes the following sections:

- About Clusters
- Preliminary Guidelines for Using Server Clusters
- Setting up a Cluster
- Adding a Server to the Server List
- Modifying Cluster Information
- Removing Servers from a Cluster
- Managing Server Clusters

About Clusters

A **cluster** ia a group of iPlanet Web Servers that can be administered from a single Administration Server. Each cluster must include one server designated as the administration server. If you have more than one cluster, you can administer all clusters from a single "master" Administration Server. The master administration server retrieves the information about all the clusters and provides the interface for managing the iPlanet Web Servers installed in their respective clusters.

Here's some of the tasks you can accomplish by organizing your servers into clusters:

- create a central place for administering all iPlanet Web Servers
- share one or more configuration files between servers
- start and stop all servers from one "master" Administration Server
- view the access and error logs for the servers you selected

By clustering your iPlanet Web Servers, you're able to specify a master Administration Server for administering all of your clusters.

Note The individual servers can be installed on any computer in a network, but the Administration Server that you designate as the "master" contains information about all clustered servers, and must have access to each cluster's individual Administration Server.

Preliminary Guidelines for Using Server Clusters

When you configure a cluster, the master Administration Server containing the information about all clusters communicates with each individual cluster's Administration Server. Because of this, each cluster-specific administration server must have an administrative user and password that the master Administration Server can use to authenticate itself. When you log in to this master administration server and you supply a username and password, that information is sent to all remote Administration Server.

Before you can create a cluster, you must first install all of the servers you want to include in the cluster. For example, if you want three clusters of five iPlanet Web Servers per cluster, you would first need to install all of the servers on the computers where they'll run, and then you would configure one of the iPlanet Web Servers in each cluster as the Administration Server, and then you'll need to configure one single cluster's administration server as the master Administration Server for all clusters. It doesn't matter which server you choose as the master administration server.

The following list provides some guidelines for configuring groups of servers into clusters:

- Install all of the servers you want to include in a particular cluster prior to creating any clusters.
- All servers in a cluster must be iPlanet Web Servers.
- Servers can be installed on any computer in a network, but the "master" Administration Server containing information about the clusters must have access to each cluster-specific Administration Server.
- Any cluster-specific Administration Server can serve as the master administration server.
- The master Administration Server retrieves information about all installed iPlanet Web Servers.
- Make sure all cluster-specific Administration Servers have a username and password that matches one used in the master administration server. You can use the distributed administration feature to set up multiple administrators on each Administration Server. For more information about distributed administration, see "Enabling Distributed Administration," on page 69 in Chapter 3, "Setting Administration Preferences."
- Make sure all Administration Server are version 4.1 and use the same protocol (HTTP or HTTPS). You'll get an error if you try to add a 3.x Enterprise Server to a cluster, and if you add a 4.1 Administration Server that has 3.x servers, the 3.x servers are not added to the clusters.
- Clusters won't work with 2.x Administration Servers.

- If you change the protocol of one Administration Server in a cluster, you must change the protocols for all Administration Servers, and then you need to update the cluster information by modifying the individual servers in the cluster.

You can perform the following tasks for working with iPlanet Web Server clusters:

- Set up a cluster
- Add servers to a cluster
- Remove servers from a cluster
- Modify cluster information
- Manage clusters

Setting up a Cluster

To set up a iPlanet Web Server cluster, perform the following steps:

1. Install the iPlanet Web Servers on the computers you want to include in the cluster. Make sure the Administration Server for the cluster has a username and password that the master Administration Server can use for authentication. You can do this either by using the default username and password or by setting up distributed administration.
2. Install the server that will contain the master Administration Server, making sure the username and password matches the one set in Step 1.
3. Add a server to the cluster list.
4. You can administer a remote server by accessing its Server Manager forms from the cluster form or by copying a configuration file from one server in the cluster to another.

Note After changing the configuration for a remote server, restart the remote server.

For more information about how to use the Server Manager forms, see "Server Manager," in the online help.

Adding a Server to the Server List

When you add a server to a cluster, you specify its Administration Server and port number. If that Administration Server contains information about more than one server, all of its servers are added to the cluster. (You can remove the individual servers later.)

Note If a remote Administration Server contains information about a cluster, the servers in the remote cluster are not added. The master Administration Server adds only those servers that are physically installed on the remote computer.

To add a remote server to the list, perform the following steps:

1. Access the Administration Server and choose the **Cluster Mgmt** tab.
2. Click the **Add Server** link.
3. Choose the protocol that the remote Administration Server uses. This is the protocol used when contacting the remote Administration Server. Choose `http` for normal Administration Server. Choose `https` if the remote Administration Server is secure.
4. Type the hostname for the remote Administration Server. If your DNS can resolve host names, you don't need to type the fully qualified domain name; otherwise type the full host and domain name. For example, type `www.mozilla.com`.
5. Type the port number that the remote Administration Server uses.
6. Click OK.

   The master Administration Server attempts to contact the remote server. When it succeeds, the server identifiers appear on the form for every server installed on the remote administration server. If you have two or more servers on different computers that use the same identifier, the form shows the server identifier and the hostname for the computer. If both server identifier and hostnames are the same, the form shows the port number. If you don't want all of the servers in the cluster, you can remove individual servers.

For more information, see "The Add Remote Servers to Cluster Database Page," in the online help.

Modifying Cluster Information

If you change an Administration Server's host name, port number, or protocol used (HTTP or HTTPS), you also need to modify the information about that Administration Server that is stored in the cluster.

To modify information about a server in a cluster, perform the following steps:

1. Go to the master Administration Server and choose the **Cluster Mgmt** tab.
2. Click the **Modify Server** link.

   All servers appear listed by their unique server identifier.
3. Check the servers you want to modify:
   1. You can change the information for all servers in the cluster by clicking **Select All**.
   2. Click **Reset Selection** to unselect any servers you have chosen in the form.
4. Choose the administration server protocol that the remote Administration Server server uses, if it has changed.
5. If applicable, type the new hostname for the remote Administration Server.
6. If applicable, type the new port number that the remote Administration Server uses.
7. Click OK.

For more information, see "The Modify Server Settings in Cluster Database Page," in the online help.

Removing Servers from a Cluster

To remove a server from the cluster, perform the following steps:

1. Go to the master Administration Server and choose the **Cluster Mgmt** tab.
2. Click the **Remove Server** link.
3. Check the server you want to remove. You can remove all servers of that type by clicking **Select All**. Click **Reset Selection** to unselect all servers.
4. Click OK.

The form displays a status saying the servers are removed from the cluster database and are no longer available for cluster control. You can still access the removed servers using their Administration Server; you just can't access them from the cluster.

For more information, see "The Remove Servers from Cluster Database Page," in the online help.

Managing Server Clusters

To manage a cluster of servers, perform the following steps:

1. Go to the Server Manager forms for the master Administration Server, and then choose the **Cluster Mgmt** tab.
2. Click the **Cluster Control** link.
3. Check the server or servers you want to change.

   Note that you can select all of the servers in the cluster by clicking **Select All**. Click **Reset Selection** to unselect any servers you have chosen in the form.
4. Configure the servers using the form elements specific to the type of server you selected. Most Netscape/iPlanet servers let you start, stop, or restart the server by clicking the corresponding buttons on the form.

For more information, see "The Cluster Control Page," in the online help.

3

Configuring and Monitoring

- **Configuring Server Preferences**
- **Understanding Log Files**
- **Using SNMP to Monitor Servers**
- **Configuring the Server for Performance**

Chapter

7

Configuring Server Preferences

This chapter describes how to configure server preferences for your iPlanet Web Server.

This chapter contains the following sections:

- Starting and Stopping the Server
- Viewing Server Settings
- Adding and Using Thread Pools
- Configuring Network Settings
- Customizing Error Responses
- Working with Dynamic Configuration Files
- Restricting Symbolic Links (Unix/Linux)
- Using the Watchdog (uxwdog) Process (Unix/Linux)

Starting and Stopping the Server

Once the server is installed, it runs constantly, listening for and accepting HTTP requests. The status of the server appears in the Server On/Off page. You can start and stop the server using one of the following methods:

- Click the Server On or Server Off in the Server On/Off page.
- Use the Services window in the Control Panel (Windows NT).
- Use `start.` If you want to use this script with `init`, you must include the start command `http:2:respawn:`*`server_root/type-identifier/`*`start -start -i` in `/etc/inittab`. (Unix/Linux)
- Use `stop`, which shuts down the server completely, interrupting service until it is restarted. If you set the `etc/inittab` file to automatically restart (using "`respawn`"), you must remove the line pertaining to the web server in `etc/inittab` before shutting down the server; otherwise, the server automatically restarts. (Unix/Linux)

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 the server is off, it 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. To change the value, add the following line to `magnus.conf`:

```
TerminateTimeout seconds
```

where *seconds* represents the number of seconds the server will wait before timing out.

The advantages to configuring this value is that the server will wait longer for connections to complete. However, because servers often have connections open from nonresponsive clients, increasing the termination timeout may increase the time it takes for the server to shut down.

Restarting the Server (Unix/Linux)

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

- Automatically restart it from the `inittab` file.

  Note that if you are using a version of Unix/Linux not derived from System V (such as SunOS 4.1.3), you will not be able to use the `inittab` file.

- Automatically restart it with daemons in `/etc/rc2.d` when the machine reboots.

- Restart it manually.

Because the installation scripts cannot edit the `/etc/rc.local` or `/etc/inittab` files, you must edit those files with a text editor. If you do not know how to edit these files, consult your system administrator or system documentation.

Normally, you cannot start an SSL-enabled server with either of these files because the server requires a password before starting. Although you can start an SSL-enabled server automatically if you keep the password in plain text in a file, this is not recommended.

Warning Leaving the SSL-enabled server's password in plain text in the server's start script is a large security risk. Anyone who can access the file has access to the SSL-enabled server's password. Consider the security risks before keeping the SSL-enabled server's password in plain text.

The server's start script, key pair file, and the key password should be owned by root (or, if a non-root user installed the server, that user account), with only the owner having read and write access to them.

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

1. Using a text editor, open the start file, which is located in *`server_root`*`/https-`*`server-id`*`.`

2. In the 10th line counting from the top of the script, insert the following:

   ```
   echo “your_SSL-enabled_server_password”|
   ```

For example, the edited line might look like this:

```
echo “MBi12!mo”|./$PRODUCT_BIN -d $PRODUCT_SUBDIR/
config $@
```

Restarting With Inittab (Unix/Linux)

To restart the server using `inittab`, put the following text on one line in the `/etc/inittab` file:

```
http:2:respawn:server_root/type-identifier/start -
start -i
```

Replace *`server_root`* with the directory where you installed the server, and replace *`type-identifier`* with the server's directory.

The `-i` option prevents the server from putting itself in a background process.

You must remove this line before you stop the server.

Restarting With the System RC Scripts (Unix/Linux)

If you use `/etc/rc.local`, or your system's equivalent, place the following line in `/etc/rc.local`:

```
server_root/type-identifier/start
```

Replace *`server_root`* with the directory where you installed the server.

Restarting the Server Manually (Unix/Linux)

To restart the server from the command line, log in as root if the server runs on ports with numbers lower than 1024; otherwise, log in as root or with the server's user account. At the command-line prompt, type the following line and press Enter:

```
server_root/type-identifier/start
```

Replace *`server_root`* with the directory where you installed the server.

You can use the optional parameters `-p` and `-i` at the end of the line:

The `-p` option starts the server on a specific port number. This overrides the setting in `magnus.conf`.

`The -i` option runs the server in `inittab` mode, so that if the server process is ever killed or crashed, `inittab` will restart the server for you. This option also prevents the server from putting itself in a background process.

Note If the server is already running, the `start` command will fail. You must stop the server first, then use the `start` command. Also, if the server startup fails, you should kill the process before trying to restart it.

Stopping the Server Manually (Unix/Linux)

If you used the `etc/inittab` file to restart the server you must remove the line starting the server from `/etc/inittab` and type `kill -1 1` before you try to stop the server. Otherwise, the server restarts automatically after it is stopped.

To stop the server manually, log in as `root` or use the server's user account (if that is how you started the server), and then type the following at the command line:

server_root`/type-identifier/stop`

Restarting the Server (Windows NT)

You can restart the server by:

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

For Windows NT 3.51, perform the following steps:

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, perform the following steps:

1. From the Start menu, choose **Settings**, and then **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 Server's User Account (Windows NT)" on page 170.

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.

Warning 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_id`*`\config`. For a default web server installation, `password.txt` would be stored in the `C:\Netscape\server4\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.

Warning 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 (Windows NT)

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 (for example, 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 (Windows NT)

To change the time interval that elapses between startup and the time the server can automatically restart, perform the following steps:

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 **Add Value** from the Edit menu. The Add Key dialog box appears.
4. In **Value Name**, 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 (Windows NT)

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, perform the following steps:

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 see if your server is running and view your server's technical and content settings. 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_id`*`\config`. For more information about the `magnus.conf` and `obj.conf` files, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

To view your server settings, see the View Server Settings page in the Server Manager.

The content settings displayed in the View Server Settings page 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.

Adding and Using Thread Pools

Use thread pools to allocate a certain number of threads to a specific service. For example, you can set up a thread pool specifically for Server-Side JavaScript applications. As part of adding the Server-Side JavaScript thread pool, you specify the maximum number of threads you want to allocate to Server-Side JavaScript applications, and they cannot take up more than their allocated number of threads.

Another use for thread pools is for running thread-unsafe plugins. By defining a pool with the maximum number of threads set to 1, only one request is allowed into the specified service function.

When you add a thread pool, the information you specify includes the minimum and maximum number of threads, the stack size, and the queue size.

The Native Thread Pool and Generic Thread Pools (Windows NT)

On Windows NT, you can use two types of thread pools: the native thread pool (`NativePool`) and additional generic thread pools.

The native thread pool is defined by default for backwards compatibility with Netscape Enterprise Server 3.6. To edit the native thread pool, see the Native Thread Pool page in the Server Manager.

You can create as many generic thread pools as you want, for as many purposes as you want. To create generic thread pools, see the Generic Thread Pools page in the Server Manager.

Thread Pools (Unix/Linux)

Since threads on Unix/Linux are always OS-scheduled (as opposed to user-scheduled) Unix/Linux users do not need to use the `NativePool`, and do not have a Server Manager page for editing its settings. However, Unix/Linux users can still create thread pools. To create thread pools, see the Thread Pools page in the Server Manager.

Editing Thread Pools

Once you have added a thread pool, you cannot change the values of the thread pool settings (minimum threads, maximum threads and so on) through the Server Manager. Instead you must edit the thread pool settings in `obj.conf`.

A thread pool appears in `obj.conf` as follows:

```
Init fn="thread-pool-init" name=name_of_the_pool
MaxThreads=n MinThreads=n QueueSize=n StackSize=n
```

Use the following parameters to change the pool: `MinThreads`, `MaxThreads`, `QueueSize`, and `StackSize`.

Windows NT users can always edit the settings for the native pool using the Server Manager.

Using Thread Pools

After you've set up a thread pool, use it by designating it as the thread pool for a specific service. For example, if you set up a thread pool you want to use for Server-Side JavaScript (SSJS) applications (this allows limiting the number of threads used for SSJS applications and isolation), you can designate it as the Sever-Side JavaScript thread pool using the Server Manager's Activate Server-Side JavaScript page.

To configure a thread pool, go to the Administration Server **Preferences** tab and select **Thread Pool**. Once a thread pool is configured, then the Javascript Thread Pool list will show the thread pool available to be used for configuring SSJS to run in that pool.

You can also designate a thread pool by using the `pool` parameter of the `load-modules` function in `obj.conf`.

```
pool="name_of_pool"
```

In addition, you can use the `pool` parameter on any NSAPI function so that only that NSAPI function runs on the pool you specify.

Configuring Network Settings

You can change the following network settings on your server: server user, server name, server port, bind to address, and MTA host.

Changing the Server's Location (Unix/Linux)

For various reasons, you might move the server from one directory to another. If you move the server, you must change the location the server references—it needs to know where the binary files are. After changing the location, you must shut down the server and copy the server files and subdirectories to a new location.

To change the server's location edit the Server Location field in the Network Settings page in the Administration Server.

Changing the Server's User Account (Unix/Linux)

The server user specifies a Unix/Linux user account that the server uses. All the server's processes run as this user.

You do not need to specify a server user if you chose a port number greater than 1024 and are not running as the `root` user (in this case, you do not need to be logged on as `root` to start the server). If you do not specify a user account here, the server runs with the user account you start it with. Make sure that when you start the server, you use the correct user account.

Note If you do not know how to create a new user on your system, contact your system administrator or consult your system documentation.

Even if you start the server as root, you should not run the server as root all the time. You want the server to have restricted access to your system resources and run as a non-privileged user. The user name you enter as the server user should already exist as a normal Unix/Linux user account. After the server starts, it runs as this user.

If you want to avoid creating a new user account, you can choose the user `nobody` or an account used by another HTTP server running on the same host. On some systems, however, the user `nobody` can own files but not run programs.

To change the server's user account, edit the Server User field in the Network Settings page in the Administration Server.

Changing the Server's User Account (Windows NT)

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, perform the following steps:

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 iPlanet Web Server 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 `iplanet.com`, you could install a server with the name `www.iplanet.com`.

If your system administrator has set up a DNS alias for your server, use that alias on the Network Settings page in the Administration Server. If you do not have a DNS alias for your server, use the machine's name combined with your domain name to construct the full hostname.

To change the server name, edit the Server Name field in The Network Settings Page in the Administration Server.

Changing the Server Port Number

The 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.iplanet.com:8090
```

Port numbers for the most commonly used network-accessible services are maintained in the file `/etc/services` (on Unix/Linux) or `\WINNT\System32\drivers\etc\services` (on Windows NT).

Although the port number can be any port from 1 to 65535, the standard insecure web server port number is 80, and the standard secure web server port number is 443.

For Unix/Linux, if you are not running as the `root` user when you install or start the server, you must use a port number higher than 1024.

To change the server port number, edit the Server Port field in the Network Settings page in the Administration Server.

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.iplanet.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, use the Bind To Address field in the Network Settings window to tell the server which IP address is associated with this hostname.

To change the server binding address, edit the Bind To Address field in the Network Settings page in the Administration Server.

Changing the Server's MTA Host

You can change the server's MTA (Message Transfer Agent) host. You must enter a valid MTA host if you want to use the agent email function.

To change the MTA Host, edit the MTA host field in the Network Settings page in the Administration Server.

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 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.

Before you can enable a custom error response, you must create the HTML file to send or the CGI program to run in response to an error. After you do this, enable the response in the Network Settings page in the Administration Server.

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 iPlanet Web Server. The subset of configuration options are stored in dynamic configuration files. Two types of dynamic configuration files are supported by iPlanet Web Server: `.htaccess` and `.nsconfig`. You can enable `.nsconfig` files in iPlanet Web Server; 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 Servers with Netscape Console*.

If you already use `.nsconfig` files, you might want to continue using them. However, iPlanet Web Server also includes a utility for converting your `.nsconfig` files to `.htaccess` files.

Using .htaccess Files

The files that support `.htaccess` are in the directory *`server_root`*`/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:

For Unix/Linux:

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

For Windows NT:

```
Init fn="load-modules" funcs="htaccess-init,htaccess-
find" \
shlib="server_root/plugins/htaccess/bin/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 must 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 iPlanet Web Server 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 Chapter 12, "Working With Configuration Styles" and Chapter 14, "Controlling Access to Your Server."

Converting Existing .nsconfig Files to .htaccess Files

The iPlanet Web 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` (The only `AuthType` supported is `Basic`.)
- `Limit`
  - order
  - deny
  - allow
  - `require`
- `AuthGroupFile`
- `AuthUserFile` (This has different formats depending on your usage. See below.)

There is an option, called `groups-with-users`, that facilitates handling large numbers of users in groups. That is, if you have many users in a group, you can follow these steps:

1. Revise the format of the user file format to list all the groups a user belongs to:

   ```
   username:password:group1,group2,group3,...groupn
   ```

2. Revise the `AuthGroupFile` directive to point to the same file as the `AuthUserFile`.

Or, alternatively, you can perform these steps:

1. Remove the `AuthGroupFile` directive entirely.

2. And add this option to the `'Init fn=htaccess-init'` line in the `obj.conf` file:

   ```
   groups-with-users="yes"
   ```

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 178.

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\server4\docs\icons\gfx\logo.gif`, instead of searching for `C:\Netscape\server4\docs\.nsconfig` the server would search all of the subdirectories:

```
C:\Netscape\server4\docs\.nsconfig
C:\Netscape\server4\docs\gfx\.nsconfig
C:\Netscape\server4\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, perform the following steps:

1. From the Server Manager, choose **Server Preferences**.
2. Click the **Dynamic Configuration Files** link.
3. Choose a resource from the Resource Picker.
4. Choose whether to base the directory from the URL or from a specified directory.
5. Type the filename.
6. Choose whether to search only the base directory.
7. Type the disabled types.
8. Click OK.
9. 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 paths 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. Note that lines are limited to a maximum of 1024 characters.

For Windows NT, 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, all of which must be lowercase. 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. 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
```

Restricting Symbolic Links (Unix/Linux)

You can limit the use of the filesystem links in your server. Filesystem links are references to files stored in other directories or filesystems. The reference makes the remote file as accessible as if it were in the current directory. There are two types of filesystem links:

- **Hard links**—A hard link is really two filenames that point to the same set of data blocks; the original file and the link are identical. For this reason, hard links cannot be on different filesystems.
- **Symbolic (soft) links**—A symbolic link consists of two files, an original file that contains the data, and another that points to the original file. Symbolic links are more flexible than hard links. Symbolic links can be used across different filesystems and can be linked to directories.

For more information about hard and symbolic links, see your Unix/Linux system documentation.

Filesystem links are an easy way to create pointers to documents outside of the primary document directory and anyone can create these links. For this reason you might be concerned that people might create pointers to sensitive files (for example, confidential documents or system password files).

To restrict symbolic links, use the Limit Symbolic Link page in the Server Manager.

Using the Watchdog (uxwdog) Process (Unix/Linux)

The `uxwdog` process is the name of the web server watchdog process, introduced in Enterprise Server 3.01. Prior to Enterprise Server 3.01, the server would fork a copy of itself at startup, and the parent server process would serve as the watchdog for the child. In Enterprise 2.x, a server restart operation would cause the parent server process (`ns-httpd`) to terminate the child `ns-httpd` process, and then recreate it. This result had the advantage that the parent process could maintain the key file password for a secure server, so that restarting the server would not require the server administrator to reenter the password.

However, with the addition of a number of subsystems to the server in Enterprise Server 3.0, it was felt that the server should be completely stopped and started for a restart operation, as the most expedient way to be sure that all subsystems were properly initialized. This had several immediate drawbacks. First, it became necessary to reenter the key file password for a secure server during a restart. This was particularly a problem for a secure server with automatic log rotation enabled, since log rotation relies on a server restart operation. Finally, every server configuration change required the server to be completely stopped and started.

The basic idea of `uxwdog` is to have a lightweight process that keeps around just enough state information to be able to start a new server process during a restart operation, without human intervention. This state consists mainly of any passwords or PINs required to start a secure server, and open file descriptors for sockets on which the server will listen. The socket file descriptors had to be kept around because some of them might be for privileged TCP ports, port 80 for example, which would require a process running as root to bind them. When this is the case, the Administration Server generally runs as root, and starts `uxwdog` as root, or else an administrator who is running as root executes the server `start` script. Once `uxwdog` binds the server listen port(s), it changes its `uid` to the server `uid`, often "nobody," and then starts the server process as that `uid`.

One consequence of this behavior is that the NSAPI Init directives always run under the server `uid`, unlike in Enterprise 3.0 and earlier, where it was possible to have them run as root. This has created some problems in upgrade situations, when a plugin Init function was creating a file during the Init. The file would be owned by root in the older server version, and when installing the plugin in Enterprise 3.01 and later, it would be necessary to change the ownership or protection on the migrated file.

In order to determine on which ports the server listens, `uxwdog` must read `magnus.conf` and `obj.conf`. It does this each time the server is restarted, and verifies that the port numbers have not been changed. If they have, a restart operation is not possible, `uxwdog` will exit, and the server will have to be manually started. This is also true if security is turned on, the server `uid` is changed, or the `PidLog` filename is changed.

The `restart` and `stop` scripts send `SIGHUP` and `SIGTERM`, respectively, to `uxwdog`. In both cases, `uxwdog` sends `SIGTERM` to the `ns-httpd` process to shut down the server. For a restart operation, `uxwdog` then creates a new server process, passing it the file descriptors of the listen ports, and any passwords or PINs it has saved.

The default behavior of the server watchdog process automatically restarts the server if the server process should terminate unexpectedly. You can revert to the previous default behavior, which was for the watchdog process to exit if the server terminates unexpectedly. To revert to the original default behavior, set the environment variable, `UXWDOG_NO_AUTOSTART`, at the beginning of the server start script as follows:

(following the "`#!/bin/sh`" line):

```
UXWDOG_NO_AUTOSTART=1; export UXWDOG_NO_AUTOSTART
```

You also now have the option to have the watchdog restart the server if the server process calls `exit()` with a non-zero argument value. This feature is disabled by default, but can be enabled by setting the `UXWDOG_RESTART_ON_EXIT` environment variable in the server start script as follows:

```
UXWDOG_RESTART_ON_EXIT=1; export
UXWDOG_RESTART_ON_EXIT
```

Between Enterprise Server 3.01 and Enterprise Server 3.5.1, the Administration Server CGIs for Enterprise Server were changed to actually restart, rather than start and stop the server, when configuration changes are applied. As part of this change, these CGIs will create a file, `wdnotify`, in the server's logs directory, which will contain a TCP port number on which the CGI listens for status from the watchdog. During a start or restart operation, `uxwdog` checks for the existence of this file, and if it finds it, connects to that port, and reads the name of a file to which `stderr` is to be redirected during the operation. uxwdog opens that file, redirects `stderr` to it, and performs the operation. If the operation is successful, `uxwdog` writes a single byte value of zero back to the CGI. Otherwise it writes a non-zero status byte, typically a value of one. Finally `uxwdog` closes the connection to the CGI, and redirects `stderr` to `/dev/console`.

There may be some cases where `wdnotify` does not get deleted when it should, which may cause `uxwdog` to exit instead of starting or restarting the server. This can be corrected by manually removing the `wdnotify` file from the `logs` directory.

Chapter

8

Understanding Log Files

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). This chapter discusses how to monitor your server by recording and viewing log files or by using the performance monitoring tools provided with your operating system.

This chapter contains the following sections:

- About Log Files
- Viewing an Access Log File
- Monitoring the Server Using HTTP
- Archiving Log Files
- Setting Log Preferences
- Flushing the Log Buffer
- Running the Log Analyzer
- Using Performance Monitor (Windows NT)
- Viewing Events (Windows NT)

About 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 `https-`*`servername`*`/logs/errors` in the server root directory, lists all the errors the server has encountered. The access log, located in `https-`*`servername`*`/logs/access` in the server root directory, records information about requests to the server and the responses from the server. You can configure the information recorded in the iPlanet Web Server `access` log file. You 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.

To view the Administration Server's access log from the Administration Server, choose the Preferences tab, and then choose the **View Access Log** page.

To view an access log from the Server Manager, choose the Status tab, and then choose the **View Access Log** page.

The following is an example of an access log in the Common Logfile Format (you specify the format in the Log Preferences window; see "Setting Log Preferences" on page 191 for more information):

```
wiley.a.com - - [16/Feb/1999:21:18:26 -0800] "GET / HTTP/1.0" 200 751
wiley.a.com - - [17/Feb/1999:1:04:38 -0800] "GET /docs/grafx/icon.gif HTTP/1.0" 204 342
wiley.a.com - - [20/Feb/1999:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
arrow.a.com - john [29/Mar/1999:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
```

Table 8.1 describes the last line of this sample access log.

Table 8.1 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)

Table 8.1 The fields in the last line of the sample access log file

Access Log Field	Example
Username	john (username entered by the client for authentication)
Date/time of request	29/Mar/1999:4:36:53 -0800
Request	GET /help
Protocol	HTTP/1.0
Status code	401
Bytes transferred	571

The following is an example of an access log using the flexible logging format (you specify the format in the Log Preferences page; see "Setting Log Preferences" on page 191 for more information):

```
wiley.a.com - - [25/Mar/1999:12:55:26 -0800] "GET /index.htm HTTP/1.0" "GET" "/?-" "HTTP/
   1.0" 304 0 - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/1999:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-" "HTTP/1.0" 304 0
   - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/1999:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-" "HTTP/1.0" 304 0
   - Mozilla/2.0 (X11; I; IRIX 5.3 IP22)
```

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. Unsuccessful user authentication is also recorded in the error log. Use the error log to find broken URL paths or missing files.

To view the Administration Server's error log file, from the Administration Server, choose the Preferences tab, and choose the **View Error Log** page.

To view a server's error log file, from the Server Manager, choose the Status tab, and choose the **View Error Log** page.

The following is an example of an error log for Unix/Linux:

```
[13/Feb/1999:16:56:51] info: successful server startup
[20/Mar/1999 19:08:52] warning: for host wiley.a.com trying to GET /report.html,
    append-trailer reports: error opening /usr/netscape/server4/docs/report.html (No such
file or directory)
```

The following is an example of an error log for Windows NT:

```
[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/Server4/docs/report.html
    (ERROR_FILE_NOT_FOUND)
```

In these examples, 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.

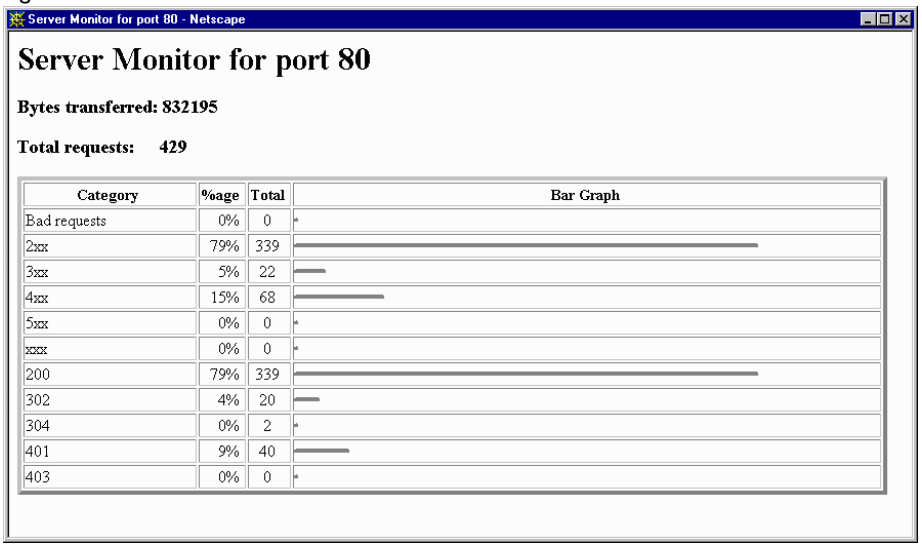
Monitoring the Server Using HTTP

You can monitor your server's usage with the interactive server monitor. You can see how many requests your server is handling and how well 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 8.1.

For a description of the various server statistics for which the interactive server monitor reports the totals, see the Monitor Current Activity page in the online help.

To monitor your server, use launch the monitoring program from the Monitor Current Activity page.

Figure 8.1 The Interactive Server Monitor



Archiving Log Files

You can set up your log files to be automatically archived. At a certain time, or after a specified interval, your logs will be rotated. iPlanet Web Server saves the old log files and stamps the saved file with a name that includes the date and time they were saved. For example, you can set up your access log files to rotate every hour, and iPlanet Web Server saves and names the file "`access.199907152400`," where name of the log file, year, month, day, and 24-hour time is concatenated together into a single character string. The exact format of the access log archive file varies depending upon which type of log rotation you set up.

iPlanet Web Server offers the two types of log rotation for archiving files: Internal-daemon log rotation and Cron-based log rotation.

Internal-daemon Log Rotation

This type of log rotation happens within the HTTP daemon, and can only be configured at startup time. Internal daemon log rotation allows the server to rotate logs internally without requiring a server restart. Logs rotated using this method are saved in the following format:

```
access.<4 digit year><2 digit month><2 digit day><4 digit 24-hour time>
```

You can specify the time used as a basis to rotate log files and start a new log file. For example, if the rotation start time is 12:00 a.m., and the rotation interval is 1440 minutes (one day), a new log file will be created immediately upon save regardless of the present time and collect information until the rotation start time. The log file will rotate every day at 12:00 a.m., and the access log will be stamped at 12:00 a.m. and saved as `access.199907152400`. Likewise, if you set the interval at 240 minutes (4 hours), the 4 hour intervals begin at 12:00 a.m. such that the access log files will contain information gathered from 12:00 a.m. to 4:00 a.m., from 4:00 a.m. to 8:00 a.m., and so forth.

If access log rotation is enabled, log file rotation starts at server startup. The first access log file to be rotated gathers information from the current time until the next rotation time. Using the previous example, if you set your start time at 12:00 a.m. and your rotation interval at 240 minutes, and the current time is 6:00 a.m., the first log file to be rotated will contain the information gathered from 6:00 a.m. to 8:00 a.m, and the next log file will contain information from 8:00 a.m. to 12:00 p.m. (noon), and so forth.

Cron-based Log Rotation

This type of log rotation is based on the time stored in the `cron.conf` file in the *`server_root`*`/https-admserv/config/` directory. This method allows you to archive log files immediately or have the server archive log files at a specific time on specific days. The server's cron configuration options are stored in `ns-cron.conf` in the *`server_root`*`/https-admserv/config/` directory. Logs rotated using the cron based method are saved as the original filename followed by the date and time the file was rotated. For example, `access` might become `access.24Apr-0430PM` when it is rotates at 4:30 p.m. For more information about cron controls, see "Using Cron Controls (Unix/Linux)" on page 75.

Log rotation is initialized at server startup. If rotation is turned on, iPlanet Web Server creates a time-stamped access log file and rotation starts at server startup.

Once the rotation starts, iPlanet Web Server creates a new time stamped log file when there is a request that needs to be logged to the access log file and it occurs after the prior-scheduled "next rotate time".

Note You should archive the server logs before running the log analyzer.

To archive log files and to specify whether to use the Internal daemon method or the cron based method, use the Archive Log Files page in the Server Manager.

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 iPlanet Web Server) what to log. A customizable format uses parameter blocks that you specify to control what gets logged. For a list of customizable format parameters, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

Once an access log for a resource has been created, you cannot change its format unless you archive it or create a new access log file for the resource.

You can specify logging preferences using The Log Preferences Page in the Server Manager, or you can manually configure the following directives in the `obj.conf` file. In `obj.conf`, the server calls the function `flex-init` to initializes the flexible logging system and the function `flex-log` to record request-specific data in a flexible log format. To log requests using the common log file format, the server calls `init-clf` to initialize the Common Log subsystem which is used if `obj.conf`, and `common-log` to record request-specific data in the common log format (used by most HTTP servers).

For more information on the NSAPI logging functions, including valid directives and parameters, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

Easy Cookie Logging

In previous versions of iPlanet Web Server, if you want to log the value of a specific cookie, you had to write a plugin API that extracted the cookie's value out of the "`Cookie`" header sent by the client, insert this value as a new variable to the request's `pblock`, and log that new variable.

iPlanet Web Server has an easy way to log a specific cookie using the flexlog facility. Add "`Req->headers.cookie.`*`cookie_name`*" to the line that initializes the `felxlog` subsystem in the configuration file `obj.conf`. This logs the value of the cookie variable *`cookie_name`* if the cookie variable is present in the request's headers, and logs "-" if it is not present.

Relaxed Logging

There is an unpleasant side effect to logging a variable other than the following standard variables: `Status`, `Content-Length`, `Client-Host`, `Full-Request`, `Method`, `Protocol`, `Query-String`, `URI`, `Referer`, `User-Agent`, `Authorization`, and `Auth-User`. Because other variables cannot be provided by the static file accelerator cache, the accelerator cache will not be used at all. Therefore performance numbers will decrease significantly for requests that would typically benefit from the accelerator, such as static files and images.

iPlanet Web Server eases the requirements of the log subsystem. Adding "`relaxed.`*`logname`*`=`*`true`*" to the "`flex-init`" line in `obj.conf` allows you to log variables outside of the standard set and still use the accelerator cache. If the accelerator is used, unavailable variables are logged as "`-`". The server does not use the accelerator for dynamic content such as CGI scrips or SHTML pages, so all the variables are always logged correctly for these requests.

Flushing the Log Buffer

You can flush the log buffer instantaneously or at a schedule other than the default time by setting buffer-flush in the `logbufInit` function in `obj.conf`. The value should be in milliseconds and greater that 0. For example:

```
int logbuf_init(pblock *args) {
   char err[MAGNUS_ERROR_LEN];
   char *t;

   logbuf_size = 0;
   logbuf_flush = DEFAULT_LOGBUF_FLUSH;
   if((t = pblock_findval("buffer-size", args)) ) {
      logbuf_size = atoi(t);
      if(logbuf_size < 0) {
         util_snprintf(err, MAGNUS_ERROR_LEN, "buffer size of %d is
invalid", logbuf_size);
         pblock_nvinsert("error", err, args);
         return REQ_ABORTED;
      }
      if(logbuf_size < (REQ_MAX_LINE * 2))
         logbuf_size = REQ_MAX_LINE * 2;
      }
      if( (t = ("buffer-flush", args)) ) {
         logbuf_flush = atoi(t) * 1000;
         if(logbuf_flush < 0) {
            util_snprintf(err, MAGNUS_ERROR_LEN,"flush rate of %d
seconds is invalid", logbuf_flush);
            pblock_nvinsert("error", err, args);
            return REQ_ABORTED;
         }
      }
   }
   return REQ_PROCEED;
}
```

Running the Log Analyzer

The *`server-install`*`/extras/log_anly` directory contains the log analysis tool that runs through the Server Manager. This log analyzer analyzes files in common log format only. The HTML document in the `log_anly` directory that explains the tool's parameters. The *`server-install`*`/extras/flex_anlg` directory contains the command-line log analyzer for the

flexible log file format. However, Server Manager defaults to using the flexible log file reporting tool, regardless of whether you've selected common or flexible log file format.

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 iPlanet Web Server or the command line.

Note Before running the log analyzer, you should archive the server logs. For more information about archiving server logs, see "Archiving Log Files" on page 189.

The following section contains instructions for running the log analyzer from the command line. To run the log analyzer from the Server Manager, see the Generate Report page in the online help.

To analyze access log files from the command line, run the tool, `flexanlg`, which is in the directory *`server-install`*`/extras/flex_anlg`.

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
```

Using Performance Monitor (Windows NT)

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 iPlanet Web Server.

To monitor iPlanet Web Server performance using Performance Monitor:

1. From the Start menu, select Programs and then Administrative Tools. Choose Performance Monitor in the Administrative Tools program group.
2. Choose **Add to Chart** from the Edit menu.

   The Add to Chart window appears.
3. If the iPlanet Web 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 monitor.

   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 errors 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 **Explain**.

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 displays a chart with your selected items. A legend at the bottom of the page displays your choices.

For more information about Performance Monitor, see the documentation for your operating system.

Viewing Events (Windows NT)

In addition to logging errors to the server error log (see "Viewing the Error Log File" on page 187), iPlanet Web Server logs severe system errors to 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. From the Start menu, select Programs and then Administrative Tools. Choose Event Viewer in the Administrative Tools program group.

2. Choose **Application** from the **Log** menu.

   The Application log appears in the Event Viewer. Errors from iPlanet Web Server has a source label of `https-`*`serverid`* or `WebServer4.1`.

3. Choose **Find** from the View menu to search for one of these labels in the log. Choose **Refresh** from the View menu to see updated log entries.

For more information about Event Viewer, consult your system documentation.

Chapter

9

Using SNMP to Monitor Servers

The majority of the content in this chapter is identical to the content in the SNMP chapter in *Managing Servers with Netscape Console*. However, some sections have been modified and new sections have been added to this chapter to make the content more specific to iPlanet Web Server.

You can use Simple Network Management Protocol (SNMP) together with Netscape/iPlanet management information bases (MIB) and network management software such as HP OpenView to monitor your servers in real-time just as you monitor other devices in your network. If you're using Windows NT, SNMP is built in and already enabled.

If you're using Unix/Linux, you must configure your Netscape/iPlanet server for SNMP if you plan to use it. This chapter provides the information you need to use SNMP on Unix/Linux with your Netscape/iPlanet server. Topics include:

- SNMP Basics
- Setting Up SNMP
- Using a Proxy SNMP Agent (Unix/Linux)
- Reconfiguring the SNMP Native Agent
- Enabling and Starting the SNMP Master Agent
- Configuring the SNMP Master Agent
- Enabling the Subagent
- Installing the SNMP Master Agent
- The iPlanet Web Server MIB

SNMP Basics

SNMP (Simple Network Management Protocol) is a protocol used to exchange data about network activity. With SNMP, data travels between a managed device and a **network management station (NMS)**, a machine users can use to remotely manage a network. A managed device is anything that runs SNMP such as hosts, routers, and Netscape/iPlanet servers.

An NMS is usually a powerful workstation with one or more network management applications installed. A network management application such as HP OpenView graphically shows information about managed devices. For example, it might show which servers in your enterprise are up or down, or the number and type of error messages received. When you use SNMP with a Netscape/iPlanet server, this information is transferred between the NMS and the sever through the use of two types of agents.

SNMP Subagent

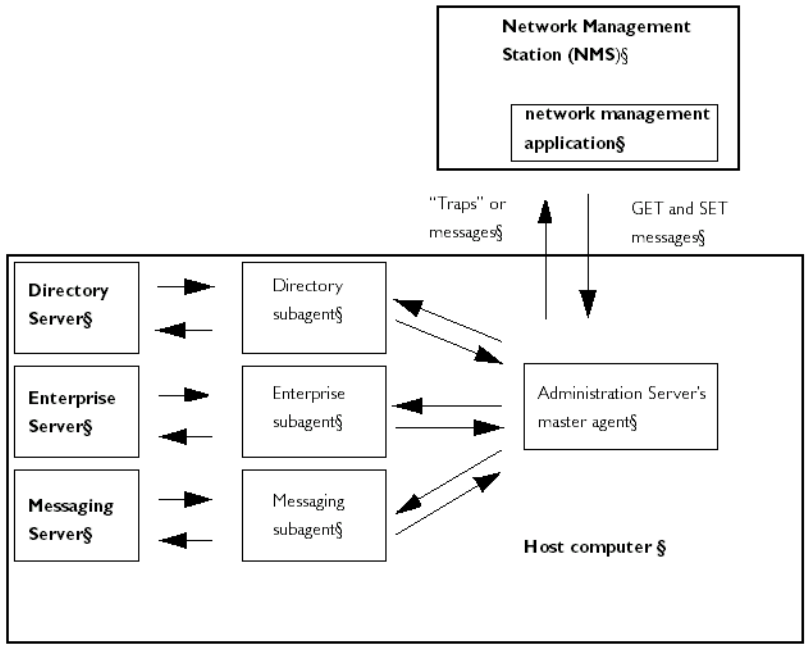
The SNMP **subagent** gathers information about the server and passes the information to the server's master agent. Every Netscape/iPlanet server, except for the Administration Server, has as subagent.

SNMP Master Agent

The **master agent** exchanges information between the various subagents and the NMS. The master agent is installed with the Administration Server.

You can have multiple subagents installed on a host computer, but only one master agent. For example, if you had Directory Server, iPlanet Web Server, and the Messaging Server installed on the same host, the subagents for each of the servers would communicate with the same master agent.

Figure 9.1 Interaction between the a network management station and a host computer.



How SNMP Works

A managed entity, such as a server, stores variables pertaining to network management. Variables that the master agent can access are known as managed objects. Managed objects are defined in a tree-like hierarchy known as a server's **management information base (MIB)**.

Each Netscape/iPlanet server subagent provides an MIB for use in SNMP communication. The MIB is a tree-like hierarchy that contains variables pertaining to the server's management. The server reports significant events to the network management station (NMS) by sending messages or **traps** containing these variables. The NMS can also query the server's MIB for data, or can remotely change variables in the MIB.

Netscape/iPlanet MIBs

Each Netscape/iPlanet server has its own management information base (MIB). All Netscape/iPlanet MIBs are located at:

```
server_root/plugins/snmp
```

A server's MIB contains variable definitions pertaining to network management for that particular server. The top level of the MIB tree is shown in Figure 9.2.

Figure 9.2 Top level of the MIB tree

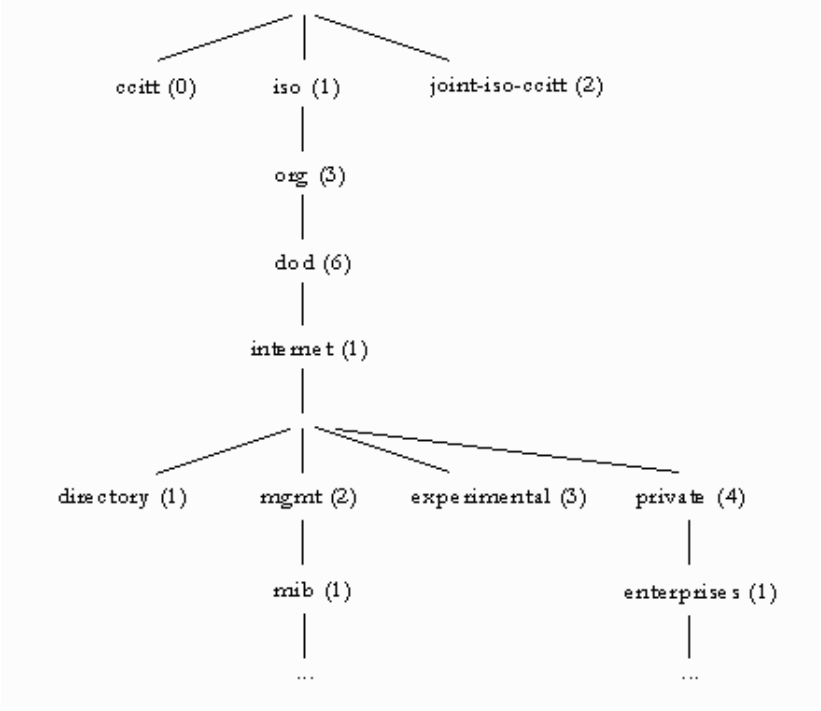


Figure 9.2 shows the internet object identifier has four subtrees: directory (1), mgmt (2), experimental (3), and private (4). The private (4) subtree contains the enterprises (1) node. Each subtree in the enterprises (1) node is assigned to an individual enterprise, which is an organization that has registered its own specific MIB extensions. An enterprise can then create product-specific subtrees

under its subtree. MIBs created by companies are located under the enterprises (1) node. The Netscape/iPlanet MIBs are located under the enterprises (1) node.

For detailed information about iPlanet Web Server's network management variables, see "The iPlanet Web Server MIB" on page 204.

The iPlanet Web Server MIB

The iPlanet Web Server MIB is a file named `netscape-http.mib`.

This file lists each object identifier for all servers currently supported. It also defines the iPlanet Web Server object identifier as *netscape 1* (`http OBJECT IDENTIFIER : := { netscape 1 }`).

Types of SNMP Messages

GET and SET are two types of messages defined by SNMP. GET and SET messages are sent by an NMS to a master agent. You can use one or the other, or both with the Administration Server. SNMP exchanges network information in the form of **protocol data units (PDUs)**. PDUs contain information about 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. PDUs sent by the server to the NMS are known as "traps." The following examples best illustrate the use of GET, SET, and "trap" messages.

NMS-initiated communication. The NMS either requests information from the server or changes the value of a variable store in the server's MIB. For example:

1. The NMS sends a message to the Administration Server master agent. The message might be a request for data (a GET message), or an instruction to set a variable in the MIB (a SET message).
2. The master agent forwards the message to the appropriate subagent.
3. The subagent retrieves the data or changes the variable in the MIB.
4. The subagent reports data or status to the master agent, and then the master agent forwards the message back (a GET message) to the NMS.

5. The NMS displays the data textually or graphically through its network management application.

Server-initiated communication. The server subagent sends a message or "trap" to the NMS when a significant event has occurred. For example:

1. The subagent informs the master agent that the server has stopped.
2. The master agent sends a message or "trap" reporting the event to the NMS.
3. The NMS displays the information textually or graphically through its network management application.

The iPlanet Web Server MIB

Each Netscape/iPlanet server has its own management information base (MIB). The iPlanet Web Server's MIB is a file called `netscape-http.mib`. This MIB contains the definitions for various variables pertaining to network management for iPlanet Web Server. These variables are known as managed objects. Using the iPlanet Web Server MIB and network management software, such as HP OpenView, you can monitor your web server like all other devices on your network.

The iPlanet Web Server MIB has an object identifier of *netscape 1* (`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 iPlanet Web Server MIB. Table 9.1 lists and describes the managed objects stored in the `netscape-http.mib`.

Table 9.1 `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/iPlanet's MIB has an object identifier of 1.3.6.1.4.1.1450).

Table 9.1 `netscape-http.mib` managed objects and descriptions

Managed object	Description
`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.
`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.

Table 9.1 `netscape-http.mib` managed objects and descriptions

Managed object	Description
`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.
`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.

Table 9.1 `netscape-http.mib` managed objects and descriptions

Managed object	Description
`httpStatisticsNum304`	The number of 304 (Not Modified) requests.
`httpStatisticsNum401`	The number of 401 (Unauthorized) requests.
`httpStatisticsNum403`	The number of 403 (Forbidden) requests.

Setting Up SNMP

In general, to use SNMP you must have a master agent and at least one subagent installed and running on a your system. You need to install the master agent before you can enable a subagent.

The procedures for setting up SNMP are different depending upon your system. Table 8.1 provides an overview of procedures you will follow for different situations. The actual procedures are described in detail later in the chapter.

Before you begin, you should verify two things:

- Is your system already running an SNMP agent (an agent native to your operating system)?
- If so, does your native SNMP agent support SMUX communication? (If you're using the AIX platform, your system supports SMUX.)

See your system documentation for information on how to verify this information.

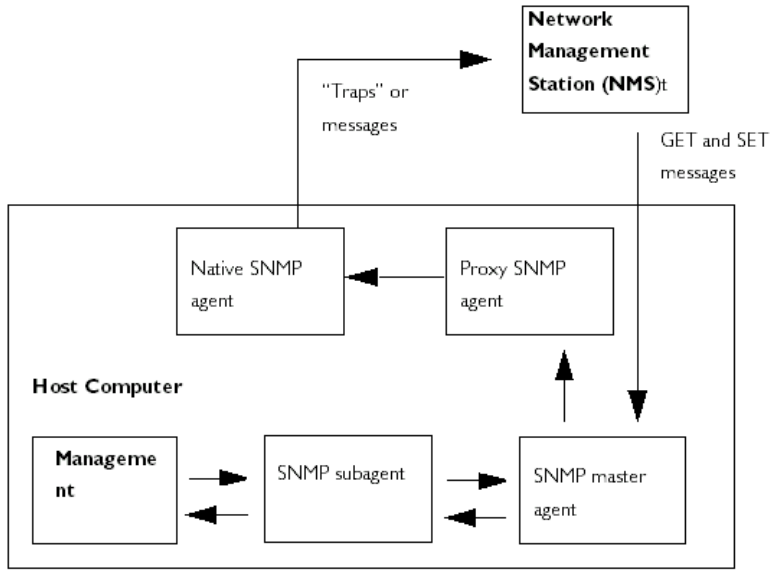
Table 9.2 Overview of procedures for enabling SNMP master agents and subagents.

If your server meets these conditions....	...follow these procedures. These are discussed in detail in the following sections.
No native agent is currently running	1. Start the master agent. 2. Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • No need to continue using native agent	1. Stop the native agent when you install the master agent for your Administration Server. 2. Start the master agent. 3. Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • Needs to continue using native agent	1. Install a proxy SNMP agent. 2. Start the proxy SNMP agent. 3. Restart the native agent using a port number other than the master agent port number. 4. Start the master agent. 5. Enable the subagent for each server installed on the system.
• Native agent is currently running • SMUX supported	1. Reconfigure the SNMP native agent. 2. Enable the subagent for each server installed on the system.

Using a Proxy SNMP Agent (Unix/Linux)

You need to use a proxy SNMP agent when you already have a native agent running, and you want to use continue using it concurrently with an iPlanet Web Server master agent. Before you start, be sure to stop the native master agent. (See your system documentation for detailed information.)

Figure 9.3 Using a proxy server when you're running a native SNMP agent.



Note To use a proxy agent, you'll need to install it and then start it. You'll also have to restart the native SNMP master agent using a port number other than the one the iPlanet Web Server master agent is running on.

Installing the Proxy SNMP Agent

If an SNMP agent is running on your system and you want to continue using the native SNMP daemon, follow the steps in these sections:

1. Install the SNMP master agent. See "Installing the SNMP Master Agent" on page 211.
2. Install and start the proxy SNMP agent and restart the native SNMP daemon. See "Using a Proxy SNMP Agent (Unix/Linux)" on page 209.
3. Start the SNMP master agent. See "Enabling and Starting the SNMP Master Agent" on page 213.
4. Enable the subagent. See "Enabling the Subagent" on page 218.

To install the SNMP proxy agent, edit the `CONFIG` file (you can give this file a different name), located in `plugins/snmp/sagt` in the server root directory, so that it includes the port that the SNMP daemon will listen to. It also needs to include the MIB trees and traps that the proxy SNMP agent will forward.

Here is an example `CONFIG` file:

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES 1.3.6.1.2.1.1,
1.3.6.1.2.1.2,
1.3.6.1.2.1.3,
1.3.6.1.2.1.4,
1.3.6.1.2.1.5,
1.3.6.1.2.1.6,
1.3.6.1.2.1.7,
1.3.6.1.2.1.8
FORWARD ALL TRAPS;
```

Starting the Proxy SNMP Agent

To start the proxy SNMP agent, at the command prompt, enter:

```
# sagt -c CONFIG&
```

Restarting the Native SNMP Daemon

After starting the proxy SNMP agent, you need to restart the native SNMP daemon at the port you specified in the `CONFIG` file. To restart the native SNMP daemon, at the command prompt, enter

```
# snmpd -P port_number
```

where *port_number* is the port number specified in the `CONFIG` file. For example, on the Solaris platform, using the port in the previously mentioned example `CONFIG` file, you'd enter:

```
# snmpd -P 1161
```

Reconfiguring the SNMP Native Agent

If your SNMP daemon is running on AIX, it supports SMUX. For this reason, you don't need to install a master agent. However, you do need to change the AIX SNMP daemon configuration.

AIX uses several configuration files to screen its communications. One of them, `snmpd.conf`, needs to be changed so that the SNMP daemon accepts the incoming messages from the SMUX subagent. For more information, see the online manual page for `snmpd.conf`. You need to add a line to define each subagent.

For example, you might add this line to the `snmpd.conf`:

```
smux 1.3.6.1.4.1.1.1450.1 "" IP_address net_mask
```

IP_address is the IP address of the host the subagent is running on, and *net_mask* is the network mask of that host.

Note Do not use the loopback address 127.0.0.1; use the real IP address instead.

Installing the SNMP Master Agent

You cannot use the Server Manager to install and start the master SNMP agent unless the server is running as `root`.

To install the master SNMP agent using the Server Manager:

1. Log in as root.
2. Check whether an SNMP daemon (`snmpd`) is running on port 161.

   If no SNMP daemon is running, go to Step 4.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports.
3. If an SNMP daemon is running, kill its process.
4. In the Server Manager, choose the **SNMP Master Agent Trap** page from the **Global Settings** tab. The Manager Entries page appears.
5. Type the name of the system that is running your network management software.
6. Type the port number at which your network management system listens for traps. (The well-known port is 162.) For more information on traps, see "Configuring Trap Destinations" on page 218.
7. Type the community string you want to use in the trap. For more information on community strings, see "Configuring the Community String" on page 217.
8. Click OK.
9. In the Server Manager, the SNMP Master Agent Community page from the choose **Global Settings** tab. The Community Strings page appears.
10. Type the community string for the master agent.
11. Choose an operation for the community.
12. Click OK.

Enabling and Starting the SNMP Master Agent

Master agent operation is defined in an agent configuration file named `CONFIG`. You can edit the `CONFIG` file using the Server Manager, or you can edit the file manually.You must install the master SNMP agent before you can enable the SNMP subagent.

If you get a bind error similar to "System Error: Could not bind to port," when restarting the master agent, use `ps -ef | grep snmp` to check if `magt` is running. If it is running, use the command `kill -9 pid` to end the process. The CGIs for SNMP will then start working again.

Manually Configuring the SNMP Master Agent

To configure the master SNMP agent manually:

1. Log in as superuser.
2. Check to see if there is an SNMP daemon (`snmpd`) running on port 161.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports. Then kill its process.
3. Edit the *CONFIG* file located in `plugins/snmp/magt` in the server root directory.
4. (Optional) Define `sysContact` and `sysLocation` variables in the `CONFIG` file.

Editing the Master Agent CONFIG File

The `CONFIG` file defines the community and the manager that master agent will work with. The manager value should be a valid system name or an IP address. Here is an example of a basic `CONFIG` file:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            manager_station_name
                   SEND ALL TRAPS TO PORT 162
                    WITH COMMUNITY public
```

Defining sysContact and sysLocation Variables

You can edit the `CONFIG` file to add initial values for `sysContact` and `sysLocation` which specify the `sysContact` and `sysLocation` MIB-II variables. The strings for `sysContact` and `sysLocation` in this example are enclosed in quotes. Any string that contains spaces, line breaks, tabs, and so on must be in quotes. You can also specify the value in hexadecimal notation.

Here is an example of a `CONFIG` file with `sysContract` and `sysLocation` variables defined:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            nms2
                   SEND ALL TRAPS TO PORT 162
                   WITH COMMUNITY public

INITIAL            sysLocation “Server room
501 East Middlefield Road
Mountain View, CA 94043
USA”

INITIAL            sysContact “John Doe
email: jdoe@netscape.com”
```

Configuring the SNMP Master Agent

To start the SNMP master agent using the Server Manager, perform the following steps:

1. Log in as root.
2. Check whether an SNMP daemon (`snmpd`) is running on port 161.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports. Then kill its process.
3. Log in to the Administration Server.
4. From the Administration Server, choose the **Global Settings** tab and click the **SNMP Master Agent Trap** link to go to the SNMP Master Agent Control page. The Manager Entries page appears. Complete the following information:

   Manager Station. Enter a valid system name or an IP address for the NMS.

   Trap Port. Enter the port number the NMS uses to listen for traps

   With Community. Enter a community string you want to use for authorization. A common default string is `public`.
5. Click the **SNMP Master Community** link. The Community Strings page appears. Enter the following community information:

   Community. Specifies the name of the community you want to use for authentication. A common default string is `public`.

   Operation. Specifies the permissions for the new community. Choose from the following:

   - *Allow All Operations*. Allows this community string to request data or reply to messages, and set variable values.
   - *Allow GET Operations*. Allow this community string to only request messages or reply to messages, and not set variables.
   - *Allow ALL Operations*. Allows this community string to only set variable values.

Starting the SNMP Master Agent

Once you have installed the SNMP master agent, you can start it manually or by using the Administration Server.

Manually Starting the SNMP Master Agent

To start the master agent manually, enter the following at the command prompt:

```
# magt CONFIG INIT&
```

The `INIT` file is a nonvolatile file that contains information from the MIB-II system group, including system location and contact information. If `INIT` doesn't already exist, starting the master agent for the first time will create it. An invalid manager name in the `CONFIG` file will cause the master agent start-up to fail.

To start a master agent on a nonstandard port, use one of two methods:

Method one: In the `CONFIG` file, specify a transport mapping for each interface over which the master agent listens for SNMP requests from managers. Transport mappings allow the master agent to accept connections at the standard port and at a nonstandard port. The master agent can also accept SNMP traffic at a nonstandard port. The maximum number of concurrent SNMP is limited by your target system's limits on the number of open sockets or file descriptors per process. Here is an example of a transport mapping entry:

```
TRANSPORT          extraordinary   SNMP
                   OVER UDP SOCKET
                    AT PORT 11161
```

After editing the `CONFIG` file manually, you should start the master agent manually by typing the following at the command prompt:

```
# magt CONFIG INIT&
```

Method two: Edit the `/etc/services` file to allow the master agent to accept connections at the standard port as well as a nonstandard port.

Starting the SNMP Master Agent Using the Administration Server

To start the SNMP master agent using the Administration Server, perform the following steps:

1. Log in to the Administration Server.
2. In the Server Manager, choose the **SNMP Master Agent Control** page from the **Global Settings** tab. The SNMP Master Agent Control page appears.
3. Click **Start**.

   You can also stop and restart the SNMP master agent from the SNMP Master Agent Control page.

Configuring the SNMP Master Agent

Once you've enabled the master agent and enabled a subagent on a host computer, you need to configure the host's Administration Server. This entails specifying community strings and trap destinations.

Configuring the Community String

A **community string** is a text string that an SNMP agent uses for authorization. This means that a network management station would send a community string with each message it sends to the agent. The agent can then verify whether the network management station is authorized to get information. Community strings are not concealed when sent in SNMP packets; strings are sent in ASCII text.

You can configure the community string for the SNMP master agent from the Community Strings page in the Server Manager. You also define which SNMP-related operations a particular community can perform. From the Server Manager, you can also view, edit, and remove the communities you have already configured.

Configuring Trap Destinations

An **SNMP trap** is a message the SNMP agent sends to a network management station. For example, an SNMP agent sends a trap when an interface's status has changed from up to down. The SNMP agent must know the address of the network management station so it knows where to send traps. You can configure this trap destination for the SNMP master agent from iPlanet Web Server. You can also view, edit, and remove the trap destinations you have already configured. When you configure trap destinations using iPlanet Web Server, you are actually editing the `CONFIG` file.

Enabling the Subagent

After you have installed the master agent that comes with the Administration Server, you must enable the subagent for your server instance before you attempt to start it. For more information on installing the master agent, see "Installing the SNMP Master Agent" on page 211. You can use the Server Manager to enable the subagent.

To stop the SNMP function on Unix/Linux platforms, you must stop the subagent first, then the master agent. If you stop the master agent first, you may not be able to stop the subagent. If that happens, restart the master agent, stop the subagent, then stop the master agent.

To enable the SNMP subagent, use the SNMP Configuration page in the Server Manager, and start the subagent from the SNMP Subagent Control page. For more information, see the corresponding sections in the online help.

Once you have enabled the subagent, you can start, stop or restart it from the SNMP Subagent Control page or the Services Control Panel for Windows NT.

Chapter

10

Configuring the Server for Performance

This chapter is intended for advanced administrators only. Be cautious when you tune your server. Do not change any values except in exceptional circumstances. Read this chapter and other relevant server documentation before making any changes. Always backup your configuration files first.

This chapter includes the following sections:

- About Server Performance
- The `perfdump` Utility
- Using `perfdump` Statistics to Tune Your Server
- Performance Buckets
- File and Accelerator Caches
- Unix/Linux Platform-Specific Issues
- Miscellaneous magnus.conf Directives
- Miscellaneous obj.conf Parameters
- Tuning the ACL Cache
- Common Performance Problems
- Sizing Issues

About Server Performance

Web servers have become increasingly important for both internal and external business communications. As web servers become more and more business-critical, server performance takes on added significance. The iPlanet Web Server, Enterprise Edition continues to lead in this area, by setting a new standard for performance.

iPlanet Web Server was designed to meet the needs of the most demanding, high traffic sites in the world. It flexibly runs on both Unix/Linux and Windows NT and can serve both static and dynamically generated content. iPlanet Web Server can also run in SSL mode, enabling the secure transfer of information.

Because iPlanet Web Server is such a flexible tool for publishing, customer needs vary significantly. This document guides you through the process of defining your server workload and sizing a system to meet your performance needs. This document addresses miscellaneous configuration and Unix/Linux platform-specific issues. It also describes the `perfdump` performance utility and tuning parameters that are built into the server. The document concludes with sizing information.

Performance Issues

The first step toward sizing your server is to determine your requirements. Performance means different things to users and to webmasters. Users want fast response times (typically less than 100 ms), high availability (no "connection refused" messages), and as much interface control as possible. Webmasters and system administrators, on the other hand, want to see high connection rates, high data throughput, and uptime approaching 100%. You need to define what performance means for your particular situation.

Here are some areas to consider:

- Peak concurrent users
- Security requirements

  Encrypting your iPlanet Web Server's data streams with SSL makes an enormous difference to your site's credibility for electronic commerce and other security-conscious applications, but it also can seriously impact your CPU's performance. Note that SSL always has a significant impact on

throughput, so minimize your use of SSL for best performance. Also, multiple CPUs help SSL, so consider buying a multi-CPU server if you need to use SSL.

- Size of document tree
- Dynamic vs. static content

  The content you serve affects your server's performance. An iPlanet Web Server delivering mostly static HTML can run much faster than a server that has to execute CGIs for every query.

Monitoring Performance

To monitor the performance of your server, use the following tools:

- The perfdump Utility
- Performance Buckets
- File Cache Dynamic Control and Monitoring

These tools are explained in more detail in the following sections of this document.

The perfdump Utility

The perfdump utility is a service function built into iPlanet Web Server that collects various pieces of performance data from the web server internal statistics and displays them in ASCII text.

To install perfdump, you need to make the following modifications in obj.conf in the netscape/server4/https-*server_name*/config directory:

1. Add the following object to your obj.conf file (after the default object):

```
<Object ppath="/usr/netscape/server4/docs/.perf">

Service fn = "service-dump"

</Object>
```

2. Edit the ppath= line if your document root is different than the example above. Make sure to put .perf after the path to the document root, as shown above.

3. Restart your server software, and access perfdump by accessing this URL:

   ```
   http://yourhost/.perf
   ```

You can request the perfdump statistics and inform the browser to automatically refresh the statistics every *n* seconds by using this URL, which sets the refresh to every 5 seconds:

```
http://yourhost/.perf?refresh=5
```

Sample Output

```
ns-httpd pid: 133
ListenSocket #0:
------------------
Address           https:\\INADDR_ANY:80
ActiveThreads     48
WaitingThreads    47
BusyThreads       1
Thread limits     48/512
KeepAliveInfo:
------------------
KeepAliveCount        0/200
KeepAliveHits         0
KeepAliveFlushes      0
KeepAliveTimeout      30 seconds
CacheInfo:
------------------
enabled             yes
CacheEntries        2/8192
CacheSize(bytes)    0/0
Hit Ratio           474254/474264 (100.00)
pollInterval        7200
Native pools:
-----------------------
```

```
NativePool:
Idle/Peak/Limit               1/1/128
Work queue length/Peak/Limit  0/0/0
Server DNS cache disabled
```

Using perfdump Statistics to Tune Your Server

This section describes the information available through the `perfdump` utility and discusses how to tune some parameters to improve your server's performance. The default tuning parameters are appropriate for all sites except those with very high volume. The only parameter that large sites may regularly need to change is the `RqThrottle` parameter, which is tunable from the Server Manager.

The `perfdump` utility monitors these statistics:

- Listen Socket Information (Listen Queue)
- KeepAlive Information
- Cache Information
- Native Thread Pools
- Asynchronous DNS Lookup (Unix/Linux)

Listen Socket Information (Listen Queue)

The listen socket is the listen queue size which is a socket-level parameter that specifies the number of incoming connections the system will accept for that socket. The default setting is 128 (for Unix/Linux) or 100 (for Windows NT) incoming connections.

Make sure your system's listen-queue size is large enough to accommodate the size set in iPlanet Web Server. The listen queue size set from iPlanet Web Server changes the listen queue size requested by your system. If iPlanet Web Server requests a listen queue size larger than your system's maximum listen queue size, the size defaults to the system's maximum.

Warning Setting the listen queue too high can degrade server performance. It 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.

The first set of `perfdump` statistics is the listen socket (or listen queue) information. For each hardware virtual server you have enabled in your server, there is one `ListenSocket` structure. For most sites, only one is listed.

```
ListenSocket #0:
------------------
Address           https:\\INADDR_ANY:80
ActiveThreads     48
WaitingThreads    47
BusyThreads       1
Thread limits     48/512
```

Note The "thread" fields specify the current thread use counts and limits for this listen socket. Keep in mind that the idea of a "thread" does not necessarily reflect the use of a thread known to the operating system. "Thread" in these fields really means an HTTP session. If you check the operating system to see how many threads are running in the process, it is not going to be the same as the numbers reported in these fields.

Tuning

Set the listen queue using the Listen Queue field in the Server Manger, using the Performance Tuning page found in the Preferences tab, or by setting the `ListenQ` parameter in `magnus.conf`.

If you are using virtual servers, there are two ways to create them: Using the `virtual.conf` file and using the `obj.conf` file. If you use the `virtual.conf` method, the 512 default maximum threads are available to all virtual servers on an as-needed basis. If you use the `obj.conf` method, the 512 threads are allocated equally to each of the defined virtual servers. For example, if you had two servers, each would have 256 threads available. This is less efficient. To maximize performance in this area, use the `virtual.conf` method.

Address

This field contains the base address that this listen socket is listening to. For most sites that are not using hardware virtual servers, the URL is:

`http://INADDR_ANY:80`"

The constant value "INADDR_ANY" is known internally to the server that specifies that this listen socket is listening on all IP addresses for this machine.

Tuning

This setting is not tunable except as described above.

ActiveThreads

The total number of "threads" (HTTP sessions) that are in any state for this listen socket. This is equal to `WaitingThreads` + `BusyThreads`. This setting is not tunable.

WaitingThreads

The number of "threads" (HTTP sessions) waiting for a new TCP connection for this listen socket.

Tuning

This is not directly tunable, but it is loosely equivalent to the `RqThrottleMinPerSocket`. See "Thread limits <min/max>" on page 226.

BusyThreads

The number of "threads" (HTTP sessions) actively processing requests which arrived on this listen socket.

This setting is not tunable.

Thread limits <min/max>

The minimum thread limit is a goal for how many threads the server attempts to keep in the WaitingThreads state. This number is just a goal. The number of actual threads in this state may go slightly above or below this value.

The maximum threads represents a hard limit for the maximum number of active threads that can run simultaneously, which can become a bottleneck for performance. iPlanet Web Server has default limits of 48/512. For more information, see "About RqThrottle (Maximum Simultaneous Connections)" on page 258.

Tuning

See "About RqThrottle (Maximum Simultaneous Connections)" on page 258.

KeepAlive Information

This section provides statistics about the server's HTTP-level KeepAlive system.

```
KeepAliveInfo:
------------------
KeepAliveCount        0/200
KeepAliveHits         0
KeepAliveFlushes      0
KeepAliveTimeout      30 seconds
```

Note The name "KeepAlive" should not be confused with TCP "KeepAlives." Also, note that the name "KeepAlive" was changed to "Persistent Connections" in HTTP/1.1, but for clarity this document continues to refer to them as "KeepAlive" connections.

Both HTTP/1.0 and HTTP/1.1 support the ability to send multiple requests across a single HTTP session. A web server can receive hundreds of new HTTP requests per second. If every request was allowed to keep the connection open indefinitely, the server could become overloaded with connections. On Unix/Linux systems, this could lead to a file table overflow very easily.

To deal with this problem, the server maintains a "Maximum number of 'waiting' keepalive connections" counter. A 'waiting' keepalive connection is a connection that has fully completed processing of the previous request over the

connection and is now waiting for a new request to arrive on the same connection. If the server has more than the maximum waiting connections open when a new connection starts to wait for a keepalive request, the server closes the oldest connection. This algorithm keeps an upper bound on the number of open, waiting keepalive connections that the server can maintain.

iPlanet Web Server does not always honor a KeepAlive request from a client. The following conditions cause the server to close a connection even if the client has requested a KeepAlive connection:

- `KeepAliveTimeout` is set to 0.
- `MaxKeepAlive` count is exceeded.
- Dynamic content, such as CGI, does not have an HTTP `content_length` header set or the header is not lowercase. This applies only to HTTP 1.0 requests. If the request is HTTP 1.1, the server honors KeepAlive requests even if the `content-length` is not set. The server now can use chunked encoding for these requests if the client can handle them (indicated by the request header `transfer-encoding: chuncked`. For more information regarding chunked encoding, see the *NSAPI Programmer's Guide*.
- Request is not `HTTP GET` or `HEAD`.
- The request was determined to be bad-- if have bad client sends only headers, no content.

KeepAliveCount <KeepAliveCount/ KeepAliveMaxCount>

The number of sessions currently waiting for a keepalive connection and the maximum number of sessions that the server allows to wait at one time.

Tuning

Edit the `MaxKeepAliveConnections` parameter in the `magnus.conf` file.

KeepAliveHits

The number of times a request was successfully received from a connection that had been kept alive.

This setting is not tunable.

KeepAliveFlushes

The number of times the server had to close a connection because the KeepAliveCount exceeded the KeepAliveMaxCount.

This setting is not tunable.

KeepAliveTimeout

Specifies the number of seconds the server will allow a client connection to remain open with no activity. A web client may keep a connection to the server open so that multiple requests to one server can be serviced by one network connection. Since a given server can handle a finite number of open connections (limited by active threads), a high number of open connections will prevent new clients from connecting.

When SSL is enabled, KeepAliveTimeout defaults to 0, which effectively disables persistent connections. If you want to use persistent connections with SSL, set KeepAliveTimeout to a non-zero value.

Tuning

You can change KeepAliveTimeout in the Performance Tuning page in the Server Manager.

Cache Information

This information applies to the accelerator cache, not the file cache. For an explanation of the caches, see "File and Accelerator Caches" on page 242.

```
CacheInfo:
------------------
enabled             yes
CacheEntries        2/8192
CacheSize(bytes)    0/0
Hit Ratio           474254/474264 (100.00)
pollInterval        7200
```

This section describes the server's cache information. The contents of a file are cached to a specific static file on disk, with the keys being the file's URI. If multiple virtual servers are set up, the key also includes the virtual server's host ID and the port number.

enabled

If the cache is disabled, the rest of this section is not displayed.

Tuning

To disable the server accelerator cache, add the following line to the `obj.conf` file:

```
Init fn=cache-init disable=true
```

CacheEntries <CurrentCacheEntries / MaxCacheEntries>

The number of current cache entries and the maximum number of cache entries. A single cache entry represents a single URI.

Tuning

To set the maximum number of cached files in the cache, add the following line to the `obj.conf` file:

```
Init fn=cache-init MaxNumberOfCachedFiles=xxxxx
```

CacheSize <CurrentCacheSize / MaxCacheSize>

The CacheSize has been deprecated for this release, since the files are cached in the file cache, not the accelerator cache. For more information, see "File and Accelerator Caches" on page 242.

Hit Ratio <CacheHits / CacheLookups (Ratio)>

The hit ratio value tells you how efficient your site is. The hit ratio should be above 90%. If the number is 0, you need to optimize your site. See the troubleshooting section for more information on how to improve your site.

If you are logging cookies and other special information, and you are seeing a very low hit rate (close to 0) refer to Chapter 8, "Understanding Log Files" for information on relaxed logging. Relaxed logging allows you to log special information and still use the accelerator cache.

This setting is not tunable.

pollInterval

Since `pollInterval` is deprecated for this release, this field displays `MaxAge` from nsfc.conf. If you have not tuned `MaxAge`, it defaults to 30 seconds.

For tuning information on this setting, see "MaxAge" on page 245.

DNS Cache Information

```
Server DNS cache disabled
```

The DNS cache caches IP addresses and DNS names.

enabled

If the cache is disabled, the rest of this section is not displayed.

Tuning

By default, the DNS cache is off. Add the following line to `obj.conf` to enable the cache:

```
Init fn=dns-cache-init
```

CacheEntries <CurrentCacheEntries / MaxCacheEntries>

The number of current cache entries and the maximum number of cache entries. A single cache entry represents a single IP address or DNS name lookup.

Tuning

To set the maximum size of the DNS cache, add the following line to the obj.conf file:

```
Init fn=dns-cache-init cache-size=xxxxx
```

HitRatio <CacheHits / CacheLookups (Ratio)>

The hit ratio is displays the number of cache hits and the number of cache lookups. A good hit ratio for the DNS cache is ~60-70%.

This setting is not tunable.

Native Thread Pools

```
Native pools:
------------------------
NativePool:
Idle/Peak/Limit               1/1/128
Work queue length/Peak/Limit  0/0/0
```

The native thread pool (NativePool) is used internally by the server to execute NSAPI functions that require a native thread for execution. Since threads on Unix/Linux are always OS-scheduled (as opposed to user-scheduled) Unix/Linux users do not need to use the native thread pool. If you want to use a thread pool on Unix/Linux, you can set one up yourself. For more information, see "Additional Thread Pools" on page 232.

iPlanet Web Server uses NSPR, which is an underlying portability layer that provides access to the host OS services. This layer provides abstractions for threads that are not always the same as those for the OS-provided threads. These non-native threads have lower scheduling overhead so their use improves performance. However, these threads are sensitive to blocking calls to the OS, such as I/O calls. To make it easier to write NSAPI extensions that can make use of blocking calls, the server keeps a pool of threads that safely support blocking calls (usually this means it is a native OS thread). During request processing, any NSAPI function that is not marked as being safe for execution on a non-native thread is scheduled for execution on one of the threads in the native thread pool.

If you have written your own NSAPI plug-ins such as NameTrans, Service, or PathCheck functions, these execute by default on a thread from the native thread pool. If your plug-in makes use of the NSAPI functions for I/O exclusively or does not use the NSAPI I/O functions at all, then it can execute on a non-native thread. For this to happen, the function must be loaded with a "NativeThread=no" option indicating that it does not require a native thread. To do this, add the following to the "load-modules" Init line in the obj.conf file:

```
Init funcs="pcheck_uri_clean_fixed_init" shlib="C:/Netscape/p186244/
P186244.dll" fn="load-modules" NativeThread="no"
```

The NativeThread flag affects all functions in the funcs list, so if you have more than one function in a library but only some of them use native threads, use separate Init lines.

Windows NT uses can edit their native thread pool settings using the Server Manager.

Additional Thread Pools

You can set up additional thread pools using the Server Manger's Preferences tab. Use thread pools to put a limit on the maximum number of requests answered by a service function at any moment. For example, these additional thread pools are a way to run thread-unsafe plugins. By defining a pool with a maximum number of threads set to 1, only one request is allowed into the specified service function.

For more information on using the Server Manager to set up additional thread pools, see the online help.

Idle/Peak/Limit

Idle indicates the number of threads that are currently idle. Peak indicates the peak number in the pool. Limit indicates the maximum number of native threads allowed in the thread pool, and is determined by the setting of NativePoolMaxThreads. For more information, see "Native Thread Pool Size" on page 234.

Tuning

Modify the NativePoolMaxThreads directive in magnus.conf, or, if you're using NT, in the Server Manager, on Native Thread Pool page under the Preferences Tab.

Work queue length/Limit

These numbers refer to a queue of server requests that are waiting for the use of a native thread from the pool. The **Work Queue Length** is the current number of requests waiting for a native thread. **Limit** is the maximum number of requests that can be queued at one time to wait for a native thread., and is determined by the setting of the NativePoolQueueSize directive in magnus.conf. For more information, see "Native Thread Pool Size" on page 234.

Tuning

Modify the NativePoolQueueSize directive in magnus.conf, or, if you're using NT, in the Server Manager, on Native Thread Pool page under the Preferences Tab.

Peak work queue length

This is the highest number of requests that were ever queued up simultaneously for the use of a native thread since the server was started. This value can be viewed as the maximum concurrency for requests requiring a native thread.

This setting is not tunable.

Work queue rejections

This is the cumulative number of requests that have needed the use of a native thread, but that have been rejected due to the work queue being full. By default, these requests are rejected with a "503 - Service Unavailable" response.

This setting is not tunable.

PostThreadsEarly

This advanced tuning parameter changes the thread allocation algorithm by causing the server to check for threads available for accept before executing a request. The default is set to Off. Recommended only in those situations when the load on the server is primarily comprised of lengthy transactions such as LiveWire and the Netscape Application Server or custom applications that access databases and other complex back-end systems. Turning this on allows the server to grow its thread pool more rapidly.

Tuning

Turn this parameter on by adding this directive to `magnus.conf`:

```
PostThreadsEarly 1
```

Native Thread Pool Size

In previous versions of the server, you controlled the native thread pool by setting system environment variables. In iPlanet Web Server, you can use the directives in `magnus.conf` to control the size of the native kernel thread pool. We recommend using the `magnus.conf` directives; however, if you have set the system environment variables previously, they override the `magnus.conf` directives.

If you are using Windows NT, you can also change these settings using the Server Manager Native Thread Pool page.

Use these directives to control the native thread pool.

`NativePoolStackSize.` Determines the stack size of each thread in the native (kernel) thread pool.

`NativePoolQueueSize.` Determines the number of threads that can wait in the queue for the thread pool. If all threads in the pool are busy, then the next request-handling thread that needs to use a thread in the native pool must wait in the queue. If the queue is full, the next request-handling thread that tries to get in the queue is rejected, with the result that it returns a busy response to the client. It is then free to handle another incoming request instead of being tied up waiting in the queue.

`NativePoolMaxThreads.` Determines the maximum number of threads in the native (kernel) thread pool.

NativePoolMinThreads. Determines the minimum number of threads in the native (kernel) thread pool.

Thread Pool Environment Variables

This section describes the thread pool environment variables. Because the Native Pool is not necessary for Unix and Linux, these environment variables default to 0 for Unix and Linux.

NSCP_POOL_WORKQUEUEMAX. This value defaults to 0x7FFFFFFF (a very large number). Setting this *below* the RqThrottle value causes the server to execute a busy function instead of the intended NSAPI function whenever the number of requests waiting for service by pool threads exceeds this value. The default returns a "503 Service Unavailable" response and logs a message if LogVerbose is enabled. Setting this *above* RqThrottle causes the server to reject connections before a busy function can execute.

This value represents the maximum number of concurrent requests for service which require a native thread. If your system is unable to fulfill requests due to load, letting more requests queue up increases the latency for requests and could result in all available request threads waiting for a native thread. In general, set this value to be high enough to avoid rejecting requests under "normal" conditions, which would be the anticipated maximum number of concurrent users who would execute requests requiring a native thread.

The difference between this value and RqThrottle is the number of requests reserved for non-native thread requests (such as static HTML, gif, and jpeg files). Keeping a reserve (and rejecting requests) ensures that your server continues to fill requests for static files, which prevents it from becoming unresponsive during periods of very heavy dynamic content load. If your server consistently rejects connections, this value is set too low or your server hardware is overloaded.

NSCP_POOL_THREADMAX. This value represents the maximum number of threads in the pool. Set this value as low as possible to sustain the optimal volume of requests. A higher value allows more requests to execute concurrently, but has more overhead due to context switching, so "bigger is not always better." If you are not saturating your CPU but you are seeing requests queue up, then increase this number. Typically, you will not need to increase this number.

Busy Functions

The default busy function returns a "503 Service Unavailable" response and logs a message if LogVerbose is enabled. You may wish to modify this behavior for your application. You can specify your own busy functions for any NSAPI function in the obj.conf file by including a service function in the configuration file in this format:

```
busy="<my-busy-function>"
```

For example, you could use this sample service function:

```
Service fn="send-cgi" busy="service-toobusy"
```

This allows different responses if the server become too busy in the course of processing a request that includes a number of types (such as Service, AddLog, and PathCheck). Note that your busy function will apply to all functions that require a native thread to execute when the default thread type is non-native.

To use your own busy function instead of the default busy function for the entire server, you can write an NSAPI init function that includes a func_insert call as shown below:

```
extern "C" NSAPI_PUBLIC int
my_custom_busy_function(pblock *pb, Session *sn,
Request *rq);

my_init(pblock *pb, Session *, Request *)

    func_insert("service-toobusy",
my_custom_busy_function);
```

Busy functions are never executed on a pool thread, so you must be careful to avoid using function calls that could cause the thread to block.

Asynchronous DNS Lookup (Unix/Linux)

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.

Enable Asynchronous DNS to avoid Multiple Thread Serialization

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 If you turn off DNS lookups on your server, host name restrictions will not work, and hostnames will not appear in your log files. Instead, you'll see IP addresses.

Caching DNS Entries

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).

Limit DNS Lookups to Asynchronous

It is recommended that you do not use DNS lookups in server processes because they are so resource intensive. If you must include DNS lookups, be sure to make them asynchronous. For more information on asynchronous DNS, see the Performance Tuning page in the online help

enabled

If asynchronous DNS is disabled, the rest of this section will not be displayed.

Tuning

Add "AsyncDNS on" to `magnus.conf`.

NameLookups

The number of name lookups (DNS name to IP address) that have been done since the server was started.

This setting is not tunable.

AddrLookups

The number of address loops (IP address to DNS name) that have been done since the server was started.

This setting is not tunable.

LookupsInProgress

The current number of lookups in progress.

This setting is not tunable.

Performance Buckets

Performance buckets allow you to define buckets, and link them to various server functions. Every time one of these functions is invoked, the server collects statistical data and adds it to the bucket. For example, `send-cgi` and `NSServletService` are functions used to serve the CGI and Java servlet requests respectively. You can either define two buckets to maintain separate counters for CGI and servlet requests, or create one bucket that counts requests for both types of dynamic content. The cost of collecting this information is little and impact on the server performance is negligible. This information can later be accessed using the `perfdump` utility. The following information is stored in a bucket:

- **Name of the bucket.** This name is used for associating the bucket with a function.
- **Description.** A description of the functions that the bucket is associated with.

- **Number of requests for this function.** The total number of times that this function was requested to be invoked.
- **Number of times the function was invoked.** This number may not coincide with the number of requests for the function because some functions may be executed more than once for a single request.
- **Function latency or the dispatch time.** The time taken by the server to invoke the function.
- **Function time.** The time spent in the function itself.

The following buckets are pre-defined by the server:

- `cache-bucket`.

  Records the statistics for accelerated cache functions. All the static content requests served using the accelerator cache are counted in this bucket.

- `default-bucket`.

  Records statistics for the functions not associated with any user defined or built-in bucket.

Configuration

Specify all the configuration information for performance buckets in the `obj.conf` file. By default the feature is disabled. To enable performance measurement add the following line in `obj.conf`:

```
Init fn="perf-init" enable=true
```

The following examples show how to define new buckets.

```
Init fn="define-perf-bucket" name="acl-bucket" description="ACL bucket"
Init fn="define-perf-bucket" name="file-bucket" description="Non-cached
responses"
Init fn="define-perf-bucket" name="cgi-bucket" description="CGI Stats"
```

The prior example creates three buckets: `acl-bucket`, `file-bucket`, and `cgi-bucket`. To associate these buckets with functions, add `bucket=`*`bucket-name`* in front of the `obj.conf` function for which you wish to measure performance.

```
PathCheck fn="check-acl" acl="default" bucket="acl-bucket"
Service method="(GET|HEAD|POST)" type="*~magnus-internal/*" fn="send-file"
bucket="file-bucket"
<Object name="cgi">
     ObjectType fn="force-type" type="magnus-internal/cgi"
     Service fn="send-cgi" bucket="cgi-bucket"
</Object>
```

Performance Report

The server statistics in buckets can be accessed using the `perfdump` utility. The performance buckets information is located in the last section of the report that `perfdump` returns. To enable reports on performance buckets, complete the following steps:

1. Define an extension for the performance bucket report. Add the following line to the `mime.types` file:

   ```
   type=perf exts=perf
   ```

2. Associate the type you declared in `mime.types` with the `service-dump` function in the `obj.conf` file:

   ```
   Service fn=service-dump type=perf
   ```

3. Use the URL `http://`*`server_name`*`:port_number/.perf` to view the performance report.

Note You must include a period (`.`) before the extension you defined in the `mime.types` file (in this case, `.perf`).

The report contains the following information:

- **Average** column shows per request statistics.
- **Request processing time** is the total time the required by the server to process all the requests it has received so far. Even on the busiest of the servers this number will be very small compared to the server uptime.
- **Number of Requests** is the total number of requests for the function.

- **Number of Invocations** is the total number that the function was invoked. This differs from the number of requests in that a function could be called multiple times while processing one request. The percentage column for this row is calculated in reference to the total number of invocations for all the buckets.
- **Latency** (in seconds) The time iPlanet Web Server takes to prepare for calling "send-cgi."
- **Function Processing Time** (in seconds) The percentage of Function Processing Time and Total Response Time is calculated with reference to the total Request processing time. (The time spent in "send-cgi" (that is, the time required to fork/exec the CGI program plus the execution time of the program itself.)
- **Total Response Time** (in seconds) The sum of Function Processing Time and Latency.
- **Percent** column displays of Number of Requests is calculated with reference to the Total number of requests.

The following is an example of the performance bucket information available through `perfdump`:

```
Performance Counters:
-----------------------------------------------
Server start time:         Mon Oct 11 15:37:26 1999
                           Average         Total      Percent

Total number of requests:                  474851
Request processing time:    0.0010      485.3198

Cache Bucket (cache-bucket)
Number of Requests:                        474254    ( 99.87%)
Number of Invocations:                     474254    ( 98.03%)
Latency:                    0.0001       48.7520    ( 10.05%)
Function Processing Time:   0.0003      142.7596    ( 29.42%)
Total Response Time:        0.0004      191.5116    ( 39.46%)

Default Bucket (default-bucket)
```

```
Number of Requests:                          597    (  0.13%)
Number of Invocations:                      9554    (  1.97%)
Latency:                    0.0000        0.1526    (  0.03%)
Function Processing Time:   0.0256      245.0459    ( 50.49%)
Total Response Time:        0.0257      245.1985    ( 50.52%)
```

File and Accelerator Caches

In iPlanet Web Server there are two caches: a front-end accelerator cache that caches response headers and contains pointers to the static file cache, and a static file cache which holds static file information as well as content. The `cache-init` directive initializes the accelerator cache. The file cache is turned on by default. If you want to change the default cache setup, you need to create a file called `nfsc.conf`. For more information, see "Configuring the File Cache" on page 244.

The file cache is implemented using a new file cache module, **NSFC**, which caches static HTML, image and sound files. In previous versions of the server, the file cache was integrated with the accelerator cache for static pages. Therefore, an HTTP request was serviced by the accelerator, or passed to the NSAPI engine for full processing, and requests that could not be accelerated did not have the benefit of file caching. This prevented many sites with NSAPI plug-ins, customized logs, or used server-parsed HTML from taking advantage of the accelerator.

The NSFC module implements an independent file cache used in the NSAPI engine to cache static files that could not be accelerated. It is also used by the accelerator cache, replacing its previously integrated file cache. NSFC caches information that is used to speed up processing of server-parsed HTML.

Configuring the Accelerator Cache

The `cache-init` function controls the accelerator caching. To optimize server speed, you should have enough RAM for the server and cache because swapping can be slow. Do not allocate a cache that is greater in size than the amount of memory on the system.

Table 10.1 The cache-init parameters

`disable`	(optional) Specifies whether the file cache is disabled or not. If set to anything but "false" the cache is disabled. By default, the cache is enabled.
`MaxNumberOfCachedFiles`	(optional) Maximum number of entries in the accelerator cache. The default is 4096, minimum is 32, maximum is 32768.
`MaxNumberOfOpenCachedFiles`	(optional) Maximum number of accel_file_cache entries with file_cache entries. Default is 512, minimum is 32, maximum is 32768.
`CacheHashSize`	(optional) size of hash table for the accelerator cache. Default is 8192, minimum is 32, maximum is 32768.
`Reaper`	(optional) Deletes old references to NSFC entries. The accelerator file cache contains references to entries in the static NSFC file cache. If set to "on", the Reaper deletes references to NSFC entries marked for deletion. Default is "on".
`ReaperInterval`	(optional) Seconds to wait before deleting old static file cache reference entries. Default is 3600 seconds. When set to 0, the reaper is disabled.
`MaxFilesToReap`	(optional) Maximum number of old static file cache reference entries to delete during each round of reaping. The default is 50. When set to 0, the reaper is disabled.
`NoOverflow`	(optional) IRIX only.
`IsGlobal`	(optional) IRIX only.

Example

```
Init fn="cache-init" MaxNumberOfCachedFiles=15000
MaxNumberOfOpenCachedFiles=15000 CacheHashSize=15101 Reaper=on
ReaperInterval=3600 MaxFilesToReap=50
```

Using the Reaper Parameters

The accelerator file cache contains references to entries in the static NSFC file cache. During the validation of the accelerator cache entry for a request, an NFSC entry may be marked for deletion if it's past its maximum age and the file has been changed. The accelerator file cache releases the old NSFC cache entry so that a new entry can be added. However, if your file cache size is not large enough, an NSFC entry may be marked for deletion to make room for a new entry. In this situation, if the entry marked for deletion has a reference in the accelerator file cache and no request for the entry ever comes again, the NSFC is not able to free the entry marked for deletion. In this situation, you can use `Reaper` to delete these entries.

Because this situation is relatively rare, set your `ReaperInterval` to an appropriately long interval, and your `MaxFilesToReap` to a small value.

Depending upon how the file and accelerator caches are configured and the load on the server, you may experience performance impact because of lock contention with request threads while reaping. Different platforms may respond to this contention differently. On some platforms (for example, Compaq Tru64 Unix 4.0d), under certain stressed configurations and heavy loads, this contention may cause more request threads to be created and hence more memory usage. In this case you need to disable `Reaper` or adjust the configuration or system resources.

Configuring the File Cache

You configure the file cache in a text file `nsfc.conf`. You can also turn off caching for a specific directory by using the parameter `nocache` in the `obj.conf` file.

Configuring nsfc.conf

By default, the file cache is turned on and uses the default values for all parameters described below. If you would like to change parameter values, you need to create a text file called `nsfc.conf` in the *`server_root`*`/https-`*`server-id`*`/config` directory. To change a parameter value for improved performance, type the parameter and its new value in the `nsfc.conf` file.

CopyFiles

When `CopyFiles` is set to "true," the file is copied to a temporary file which the server displays when users access the file. Defaults to "false" on Unix/Linux and "true" on Windows NT. The temporary file is stored in the directory specified in `TempDir`.

```
CopyFiles=true
```

FileCacheEnable

Specifies whether the file cache is enabled or not. By default, this is set to true.

```
FileCacheEnable=true
```

HashInitSize

The size of the file cache hash table. The default size is 2 * MaxFiles +1. For example, if your `MaxFiles` is set to 1024, the default `HashInitSize` is 2049.

```
HashInitSize=9131
```

HitOrder

If the `MaxFiles` limit has been reached when the server creates a new file cache entry, the server marks an existing entry for deletion. If `HitOrder` is set to "true" the file entry marked for deletion is the one that has received the fewest hits. If `HitOrder` is set to "false" the file entry marked for deletion is the one that has gone the longest without a hit.

```
HitOrder=true
```

MaxAge

The maximum age (in seconds) of a valid cache entry. This setting controls how long cached information will continue to be used once a file has been cached. An entry older than `MaxAge` is replaced by a new entry for the same file if the same file is referenced through the cache.

Set `MaxAge` based on whether the content is updated (existing files are modified) on a regular schedule or not. For example, if content is updated four times a day at regular intervals, `MaxAge` could be set to 21600 seconds (6 hours). Otherwise, consider setting `MaxAge` to the longest time you are willing to serve the previous version of a content file, after the file has been modified.

By default, this is set to 30.

```
MaxAge=30
```

MaxFiles

The maximum number of files that may be in the cache at once.

By default, this is set to 1024.

```
MaxFiles=1024
```

MediumFileSizeLimit (Unix/Linux)

The size (in bytes) of the largest file that is not a "small" file that is considered to be "medium" size. The contents of medium files are cached by mapping the file into virtual memory (currently only on Unix/Linux platforms). The contents of "large" files (larger than "medium") are not cached, although information about large files is cached.

By default, this is set to 525000 (525 KB).

```
MediumFileSizeLimit=525000
```

MediumFileSpace

The size (in bytes) of the virtual memory used to map all medium sized files.

By default, this is set to 10000000 (10MB).

```
MediumFileSpace=10000000
```

SmallFileSizeLimit (Unix/Linux)

The size (in bytes) of the largest file considered to be "small." The contents of "small" files are cached by allocating heap space and reading the file into it.

The idea of distinguishing between small files and medium files is to avoid wasting part of many pages of virtual memory when there are lots of small files. So the `SmallFileSizeLimit` would typically be a slightly lower value than the VM page size.

By default, this is set to 2048.

```
SmallFileSizeLimit=2048
```

SmallFileSpace

The size of heap space (in bytes) used for the cache, including heap space used to cache small files.

By default, this is set to 1MB for Unix/Linux, 0 for Windows NT.

```
SmallFileSpace=1000000
```

TempDir

`TempDir` sets the directory name where the temporary files are copied if `CopyFiles` is set to "true." Defaults to s*ystem_temp_dir*`/netscape/`*`server_instance`*.

If you assign a temporary directory, the server creates a structure within that directory for the temporary files. For example, on Windows NT if you set the temporary directory to C:/mytemp, the temporary files are created in the file `C:/mytemp/c/server_doc_root`. The `c` directory comes from the drive letter.

```
TempDir=C:/temp
```

TransmitFile

When `TransmitFile` is set to "true," open file descriptors are cached for files in the file cache, rather than the file contents, and `PR_TransmitFile` is used to send the file contents to a client. When set to "true," the distinction normally made by the file cache between small, medium, and large files no longer applies, since only the open file descriptor is being cached. By default, `TransmitFile` is "false" on Unix/Linux and "true" on Windows NT.

This directive is intended to be used on Unix/Linux platforms that have native OS support for `PR_TransmitFile`, which currently includes HPUX and AIX. It is not recommended for other Unix/Linux platforms.

```
TransmitFile=true
```

Using the nocache Parameter

You can use the parameter `nocache` for the Service function `send-file` to specify that files in a certain directory not be cached. For example, if you have a set of files that changes too rapidly for caching to be useful, you can put them in a directory and instruct the server not to cache files in that directory.

For example:

```
<Object name=default>
...
NameTrans fn="pfx2dir" from="/myurl" dir="/export/mydir", name="myname"
...
Service method=(GET|HEAD|POST) type=*~magnus-internal/* fn=send-file
...
</Object>
<Object name="myname">
Service method=(GET|HEAD) type=*~magnus-internal/* fn=send-file nocache=""
</Object>
```

In the above example, the server does not cache static files from `/export/mydir/` when requested by the URL prefix `/myurl`

File Cache Dynamic Control and Monitoring

An object can be added to `obj.conf` to enable the NSFC file cache to be dynamically monitored and controlled while the server is running. Typically this would be done by first adding a `NameTrans` directive to the "default" object:

```
NameTrans fn="assign-name" from="/nsfc" name="nsfc"
```

Then add a new object definition:

```
<Object name=”nsfc”>
     Service fn=service-nsfc-dump
</Object>
```

This enables the file cache control and monitoring function (`nsfc-dump`) to be accessed via the URI, "`/nsfc`." By changing the "from" parameter in the `NameTrans` directive, a different URI can be used.

The following is an example of the information you receive when you access the URI:

```
iPlanet Web Server File Cache Status (pid 7960)

The file cache is enabled.

Cache resource utilization

Number of cached file entries = 1039 (112 bytes each, 116368 total bytes)
Heap space used for cache = 237641/1204228 bytes
Mapped memory used for medium file contents = 5742797/10485760 bytes
Number of cache lookup hits = 435877/720427 ( 60.50 %)
Number of hits/misses on cached file info = 212125/128556
Number of hits/misses on cached file content = 19426/502284
Number of outdated cache entries deleted = 0
Number of cache entry replacements = 127405
Total number of cache entries deleted = 127407
Number of busy deleted cache entries = 17

Parameter settings

HitOrder: false
CacheFileInfo: true
CacheFileContent: true
TransmitFile: false
MaxAge: 30 seconds
MaxFiles: 1024 files
SmallFileSizeLimit: 2048 bytes
MediumFileSizeLimit: 537600 bytes

CopyFiles: false
Directory for temporary files: /tmp/netscape/https-axilla.mcom.com
Hash table size: 2049 buckets
```

You can include a query string = when you access the "`/nsfc`" URI. The following values are recognized:

- `?list` - Lists the files in the cache.
- `?refresh=`*n* - Causes the client to reload the page every *n* seconds.
- `?restart` - Causes the cache to be shut down and then started.
- `?start` - Starts the cache.
- `?stop` - Shuts down the cache.

If you choose the `?list` option, the file listing includes the file name, a set of flags, the current number of references to the cache entry, the size of the file, and an internal file ID value. The flags are as follows:

- C - File contents are cached.
- D - Cache entry is marked for delete.
- E - `PR_GetFileInfo()` returned an error for this file.
- I - File information (size, modify date, etc.) is cached.
- M - File contents are mapped into virtual memory.
- O - File descriptor is cached (when TransmitFile is set to true).
- P - File has associated private data (should appear on `shtml` files).
- T - Cache entry has a temporary file.
- W - Cache entry is locked for write access.

For sites with scheduled updates to content, consider shutting down the cache while the content is being updated, and starting it again after the update is complete. Although performance will slow down, the server operates normally when the cache is off.

Unix/Linux Platform-Specific Issues

The various Unix/Linux platforms all have limits on the number of files that can be open in a single process at one time. For busy sites, increase that number to 1024.

- Solaris: in `/etc/system`, set `rlim_fd_max`, and `reboot`.
- AIX: run `smit` and check the kernel tuning parameters.
- HP-UX: run `sam` and check the kernel tuning parameters.

These Unix platforms have proprietary sites for additional information about tuning their systems for web servers:

- AIX - `http://www.rs6000.ibm.com/resource/technology/sizing.html`
- IRIX - `http://www.sgi.com/tech/web/`
- Compaq Tru64 Unix - `http://www.unix.digital.com/internet/tuning.htm`
- SUN - `http://www.sun.com/sun-on-net/performance/book2ref.html`

Tuning Solaris for Performance Benchmarking

The following table shows the operating system tuning for Solaris used when benchmarking for performance and scalability. These values are an example of how you might tune your system to achieve the desired result.

Table 10.2 Tuning Solaris for performance benchmarking

Parameter	Scope	Default Value	Tuned Value	Comments
rlim_fd_max	/etc/system	1024	8192	Process open file descriptors limit; should account for the expected load (for the associated sockets, files, pipes if any).
rlim_fd_cur	/etc/system	64	8192	
sq_max_size	/etc/system	2	0	Controls streams driver queue size; setting to 0 makes it infinity so the performance runs wont be hit by lack of buffer space. Set on clients too.
tcp_close_wait_interval	ndd /dev/tcp	240000	60000	Set on clients too.
tcp_time_wait_interval	ndd /dev/tcp	240000	60000	For Solaris 7 only. Set on clients too.
tcp_conn_req_max_q	ndd /dev/tcp	128	1024	
tcp_conn_req_max_q0	ndd /dev/tcp	1024	4096	
tcp_ip_abort_interval	ndd /dev/tcp	480000	60000	
tcp_keepalive_interval	ndd /dev/tcp	7200000	900000	For high traffic web sites lower this value.
tcp_rexmit_interval_initial	ndd /dev/tcp	3000	3000	If retransmission is greater than 30-40%, you should increase this value.
tcp_rexmit_interval_max	ndd /dev/tcp	240000	10000	
tcp_rexmit_interval_min	ndd /dev/tcp	200	3000	
tcp_smallest_anon_port	ndd /dev/tcp	32768	1024	Set on clients too.
tcp_slow_start_initial	ndd /dev/tcp	1	2	Slightly faster transmission of small amounts of data.
tcp_xmit_hiwat	ndd /dev/tcp	8129	32768	To increase the transmit buffer.
tcp_recv_hiwat	ndd /dev/tcp	8129	32768	To increase the receive buffer.

Tuning HP-UX for Performance Benchmarking

The following table shows the operating system tuning for HP-UX used when benchmarking for performance and scalability. These values are an example of how you might tune your system to achieve the desired result.

Table 10.3 Tuning HP-UX for performance benchmarking

Parameter	Scope	Default Value	Tuned Value	Comments
maxfiles	/stand/ system	2048	4096	Must edit file by hand to increase beyond 2048 limit allowed by sam
maxfiles_lim	/stand/ system	2048	4096	Must edit file by hand to increase beyond 2048 limit allowed by sam
tcp_time_wait_interval	ndd /dev/tcp	60000	60000	
tcp_conn_req_max	ndd /dev/tcp	20	1024	
tcp_ip_abort_interval	ndd /dev/tcp	600000	60000	
tcp_keepalive_interval	ndd /dev/tcp	72000000	900000	
tcp_rexmit_interval_initial	ndd /dev/tcp	1500	1500	
tcp_rexmit_interval_max	ndd /dev/tcp	60000	60000	
tcp_rexmit_interval_min	ndd /dev/tcp	500	500	
tcp_xmit_hiwater_def	ndd /dev/tcp	32768	32768	
tcp_recv_hiwater_def	ndd /dev/tcp	32768	32768	

Miscellaneous magnus.conf Directives

You can use the following `magnus.conf` directives to configure your server to function more effectively:

- `RqThrottle` (See "About RqThrottle (Maximum Simultaneous Connections)" on page 258)
- `PostThreadsEarly` (See "PostThreadsEarly" on page 234)

- `MaxProcs` (See "Multi-process Mode" on page 254)
- `MinAcceptThreadPerSocket` and `MaxAcceptThreadPerSocket` (See"Accept Thread Information" on page 256)
- `AcceptTimout` (See "Accept Timeout Information" on page 256)
- `MinCGIStubs`, `MaxCGIStubs`, and `CGIStubIdleTimeout` (See "CGIStub Processes (Unix/Linux)" on page 256)
- `SndBufSize` and `RcvBufSize` (See "Buffer Size" on page 257)
- StrictHttpHeader (See "Strict HTTP Header Checking" on page 258)

Multi-process Mode

You can configure the server to handle requests using multiple processes and multiple threads in each process. This flexibility provides optimal performance for sites using threads and also provides backward compatibility to sites running legacy applications that are not ready to run in a threaded environment. Because applications on Windows NT generally already take advantage of multi-process considerations, this feature mostly applies to Unix/Linux platforms.

The advantage of multiple processes is that legacy applications which are not thread-aware or thread safe can be run more effectively in iPlanet Web Server. However, because all the Netscape/iPlanet extensions are built to support a single-process, threaded environment, they cannot run in the multi-process mode. WAI, LiveWire, Java, Server-side JavaScript, LiveConnect and the Web Publishing and Search plug-ins fail on startup if the server is in multi-process mode.

You can run your iPlanet Web Server in one of the following two modes:

- iPlanet Web Server with a single process
- iPlanet Web Server with multiple processes

In the single-process mode, the server receives requests from web clients to a single process. Inside the single server process, many threads are running which are waiting for new requests to arrive. When a request arrives, it is handled by the thread receiving the request. Because the server is multi-threaded, all extensions written to the server (NSAPI) must be thread-safe. This means that if the NSAPI extension uses a global resource (like a shared reference to a file or global variable) then the use of that resource must be synchronized so that only one thread accesses it at a time. All plug-ins provided

by Netscape/iPlanet are thread-safe and thread-aware, providing good scalability and concurrency. However, your legacy applications may be single-threaded. When the server runs the application, it can only execute one at a time. This leads to severe performance problems when put under load. Unfortunately, in the single-process design, there is no real workaround.

In the multi-process design, the server spawns multiple server processes at startup. Each process contains one or more threads (depending on the configuration) which receive incoming requests. Since each process is completely independent, each one has its own copies of global variables, caches, and other resources. Using multiple processes requires more resources from your system. Also, if you try to install an application which requires shared state, it has to synchronize that state across multiple processes. NSAPI provides no helper functions for implementing cross-process synchronization.

If you are not running any NSAPI in your server, you should use the default settings: one process and many threads. If you are running an application which is not scalable in a threaded environment, you should use a few processes and many threads, for example, 4 or 8 processes and 256 or 512 threads per process.

`MaxProcs` (Unix/Linux)

Use this directive to set your Unix/Linux server in multi-process mode, which allows for higher scalability on multi-processor machines. If, for example, you are running on a four-processor CPU, setting `MaxProcs` to `4` improves performance: one process per processor.

If you are running iPlanet Web Server in multi-process mode, you cannot run LiveWire, Web Publisher, and WAI.

This directive results in one primordial process and four active processes:

```
MaxProcs 4
```

Note This value is not tunable from the Server Manager. You must use magnus.conf.

Accept Thread Information

`MinAcceptThreadsPerSocket` / `MaxAcceptThreadsPerSocket`

Use these directives to specify how many threads you want in accept mode on a listen socket at any time. It's a good practice to set this to equal the number of processes. You can set this to twice (`2x`) the number of processes, but setting it to a number that is too great (such as ten (`10x`) or fifty (50x)) allows too many threads to be created and slows the server down.

Accept Timeout Information

AcceptTimeout

Use this directive to specify the number of seconds the server waits between accepting a connection to a client and receiving information from it. The default setting is 30 seconds. Under most circumstances you should not have to change this setting. By setting it to less than the default 30 seconds, you can free up threads sooner. However, you may also disconnect users with slower connections.

CGIStub Processes (Unix/Linux)

You can adjust the `CGIStub` parameters on Unix/Linux systems. In the iPlanet Web Server, the CGI engine creates `CGIStub` processes as needed to handle CGI processes. On systems that serve a large load and rely heavily on CGI-generated content, it is possible for the `CGIStub` processes spawned to consume all system resources. If this is happening on your server, the `CGIStub` processes can be tuned to restrict how many new `CGIStub` processes can be spawned, their timeout value, and the minimum number of `CGIStub` processes that will be running at any given moment.

Note If you have an `init-cgi` function in the `obj.conf` file and you are running in multi-process mode, you must add `LateInit = yes` to the `init-cgi` line.

MinCGIStubs/MaxCGIStubs/CGIStubIdleTimeout

The three directives (and their defaults) that can be placed in the `magnus.conf` file to control `Cgistub` are:

```
MinCGIStubs             2
MaxCGIStubs            10
CGIStubIdleTimeout     45
```

`MinCGIStubs` controls the number of processes that are started by default. The first `CGIStub` process is not started until a CGI program has been accessed. The default value is 2. Note that if you have a `init-cgi` directive in the `obj.conf` file, the minimum number of `CGIStub` processes are spawned at startup.

`MaxCGIStubs` controls the maximum number of `CGIStub` processes the server can spawn. This is the maximum *concurrent* `CGIStub` processes in execution, not the maximum number of pending requests. The default value shown should be adequate for most systems. Setting this too high may actually reduce throughput. The default value is 10.

`CGIStubIdleTimeout` causes the server to kill any `CGIStub` processes that have been idle for the number of seconds set by this directive. Once the number of processes is at `MinCGIStubs` it does not kill any more processes.

Buffer Size

SndBufSize/RcvBufSize

You can specify the size of the send buffer (`SndBufSize`) and the receiving buffer (`RcvBufSize`) at the server's sockets. For more information regarding these buffers, see your Unix/Linux documentation.

Strict HTTP Header Checking

StrictHttpHeaders

The server provides strict HTTP header checking, rejecting connections that include inappropriately duplicated headers. If you want to suppress this check, you can turn strict header checking off by setting the `StrictHttpHeaders` directive to `off` in `magnus.conf`:

```
StrictHttpHeaders off
```

About RqThrottle (Maximum Simultaneous Connections)

The `RqThrottle` parameter in the `magnus.conf` file specifies the maximum number of simultaneous transactions the web server can handle. The default value is 512. Changes to this value can be used to throttle the server, minimizing latencies for the transactions that are performed. The `RqThrottle` value acts across multiple virtual servers, but does not attempt to load-balance.

To compute the number of simultaneous requests, the server counts the number of active requests, adding one to the number when a new request arrives, subtracting one 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 for static files 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. (You can define this timeout period in `obj.conf`. It has a default of 5 minutes.) 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.

The defaults are 48/512. If your site is experiencing slowness and the `ActiveThreads` count remains close to the limit, consider increasing the maximum threads limit. To find out the active thread count, use the `perfdump` utility.

A suitable `RqThrottle` value ranges from 200-2000 depending on the load. If you want your server to use all the available resources on the system (that is, you don't run other server software on the same machine), then you can increase `RqThrottle` to a larger value than necessary without negative consequences.

Note If you are using older NSAPI plug-ins that are not reentrant, they will not work with the multithreading model described in this document. To continue using them, you should revise them so that they are reentrant. If this is not possible, you can configure your server to work with them by setting `RqThrottle` to 1 and then using a high value for `MaxProcs`, such as 48 or greater, but this will adversely impact your server's performance.

Tuning

There are two ways to tune the thread limit: through editing the `magnus.conf` file and through the Server Manager.

If you edit the `magnus.conf` file, `RqThrottleMinPerSocket` is the minimum value and `RqThrottle` is the maximum value.

The minimum limit is a goal for how many threads the server attempts to keep in the `WaitingThreads` state. This number is just a goal. The number of actual threads in this state may go slightly above or below this value. The default value is 48. The maximum threads represents a hard limit for the maximum number of active threads that can run simultaneously, which can become a bottleneck for performance. The default value is 512.

If you use the Server Manager, follow these steps:

1. Go to the **Preferences** tab.
2. Click the **Performance Tuning** link.

3. Enter the desired value in the **Maximum simultaneous requests** field.

For additional information, see the online help for the Performance Tuning page.

Miscellaneous obj.conf Parameters

You can use some obj.conf function parameters to improve your server's performance. In addition to the ones listed below, see "Using the nocache Parameter" on page 248 for information on that parameter.

For more information on using `obj.conf`, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

find-pathinfo-forward

The parameter `find-pathinfo-forward` for the `PathCheck` function `find-pathinfo` and the `NameTrans` functions `pfx2dir` and `assign-name` can help you improve your performance. This parameter instructs the server to look for `PATH_INFO` forward in the path after `ntrans-base` instead of backward from the end of path in the server function `find-pathinfo`.

Note The server ignores the `find-pathinfo-forward` parameter if the `ntrans-base` parameter is not set in `rq->vars` when the server function `find-pathinfo` is called. By default, `ntrans-base` is set.

For example:

```
NameTrans fn="pfx2dir" find-pathinfo-forward=""
from="/cgi-bin" dir="/export/home/cgi-bin" name="cgi"

NameTrans fn="assign-name" from="/perf" find-pathinfo-
forward="" name="perf"
```

This feature can improve performance for certain URLs by doing fewer stats in the server function `find-pathinfo`. On Windows NT, you can also use this feature to prevent the server from changing "\" to "/" when using the `PathCheck` server function `find-pathinfo`.

nostat

You can specify the parameter `nostat` in the `NameTrans` function `assign-name` to prevent the server from doing a stat on a specified URL whenever possible. Use the following syntax:

```
nostat=virtual-path
```

For example:

```
<Object name=default>

...

NameTrans fn="assign-name" from="/nsfc"  nostat="/
nsfc" name="nsfc"

...

</Object>

<Object name=nsfc>

     Service fn=service-nsfc-dump

     </Object>
```

In the above example, the server does not stat for path */ntrans-base*/nsfc and */ntrans-base*/nsfc/* if */ntrans-base* is set. If */ntrans-base* is not set, the server does not stat for URLs /nsfc and /nsfc/*. By default */ntrans-base* is set. The example assumes the default `PathCheck` server functions are used.

When you use `nostat=`*`virtual-path`* in the `assign-name` `NameTrans`, the server assumes that stat on the specified */virtual-path* will fail. Therefore, use `nostat` only when the path of the */virtual-path* does not exist on the system, for example, in NSAPI plugin urls. Using nostat on those URLs improves performance by avoiding unnecessary stats on those URLs.

Tuning the ACL Cache

Because of the default size of the cache (200 entries), the ACL cache can be a bottleneck or can simply not serve its purpose on a heavily trafficked site. On a heavily trafficked site well more than 200 users can hit ACL-protected resources in less time than the lifetime of the cache entries. When this situation occurs, the iPlanet Web Server has to query the LDAP server more often to validate users, which impacts performance.

This bottleneck can be avoided by increasing the size of the ACL cache with the `ACLUserCacheSize` directive in `magnus.conf`. Note that increasing the cache size will use more resources; the larger you make the cache the more RAM you'll need to hold it.

There can also be a potential (but much harder to hit) bottleneck with the number of groups stored in a cache entry (by default four). If a user belongs to five groups and hits five ACLs that check for these different groups within the ACL cache lifetime, an additional cache entry is created to hold the additional group entry. When there are two cache entries, the entry with the original group information is ignored.

While it would be extremely unusual to hit this possible performance problem, the number of groups cached in a single ACL cache entry can be tuned with the `ACLGroupCacheSize` directive.

Using magnus.conf Directives

In order to adjust the cache values you will need to manually add these directives to your `magnus.conf` file.

ACLCacheLifetime

Set this directive to a number that determines the number of seconds before the cache entries expire. Each time an entry in the cache is referenced, its age is calculated and checked against `ACLCacheLifetime`. The entry is not used if its age is greater than or equal to the `ACLCacheLifetime`. The default value is 120 seconds. If this value is set to 0, the cache is turned off. If you use a large number for this value, you may need to restart the iPlanet Web Server when you make changes to the LDAP entries. For example, if this value is set to 120

seconds, the iPlanet Web Server might be out of sync with the LDAP server for as long as two minutes. If your LDAP is not likely to change often, use a large number.

ACLUserCacheSize

Set this directive to a number that determines the size of the User Cache (default is 200).

ACLGroupCacheSize

Set this directive to a number that determines how many group IDs can be cached for a single UID/cache entry (default is 4).

Verifying Settings Using LogVerbose

If you set `LogVerbose` to on, you can verify that the ACL cache settings are being used. When `LogVerbose` is running you should expect to see these messages in your errors log when the server starts:

```
User authentication cache entries expire in ### seconds.

User authentication cache holds ### users.

Up to ### groups are cached for each cached user.
```

Warning Do not turn on `LogVerbose` on a production server, because doing so degrades performance and increases the size of your error logs considerably.

Common Performance Problems

This section discusses a few common performance problems to check for on your web site:

- Low-Memory Situations
- Under-Throttled Server
- Cache Not Utilized

- KeepAlive Connections Flushed
- Log File Modes
- Using Local Variables

Low-Memory Situations

If you need iPlanet Web Server to run in low-memory situations, try reducing the thread limit to a bare minimum by lowering the value of `RqThrottle` in your `magnus.conf` file. Also you may want to reduce the maximum number of processes that the iPlanet Web Server will spawn by lowering the value of the `MaxProcs` value in the `magnus.conf` file.

Under-Throttled Server

The server does not allow the number of active threads to exceed the Thread Limit value. If the number of simultaneous requests reaches that limit, the server stops servicing new connections until the old connections are freed up. This can lead to increased response time.

In iPlanet Web Server, the server's default `RqThrottle` value is 512. If you want your server to accept more connections, you need to increase the `RqThrottle` value.

Checking

The symptom of an under-throttled server is a server with a long response time. Making a request from a browser establishes a connection fairly quickly to the server, but on under-throttled servers it may take a long time before the response comes back to the client.

The best way to tell if your server is being throttled is to look at the `WaitingThreads` count. If this number is getting close to 0 or is 0, then the server is not accepting new connections right now. Also check to see if the number of `ActiveThreads` and `BusyThreads` are close to their limits. If so, the server is probably limiting itself.

Tuning

See “About RqThrottle (Maximum Simultaneous Connections)” on page 258.

Cache Not Utilized

If the cache is not utilized, your server is not performing optimally. Since most sites have lots of GIF or JPEG files (which should always be cacheable), you need to use your cache effectively.

Some sites, however, do almost everything through CGIs, SHTML, or other dynamic sources. Dynamic content is generally not cacheable and inherently yields a low cache hit rate. Don’t be too alarmed if your site has a low cache hit rate. The most important thing is that your response time is low. You can have a very low cache hit rate and still have very good response time. As long as your response time is good, you may not care that the cache hit rate is low.

Checking

Begin by checking your Hit Ratio. This is the percentage of times the cache was used with all hits to your server. A good cache hit rate is anything above 50%. Some sites may even achieve 98% or higher.

In addition, if you are doing a lot of CGI or NSAPI calls, you may have a low cache hit rate.

Tuning

If you have custom NSAPI functions (`nametrans`, `pathcheck`, etc), you may have a low cache hit rate. If you are writing your own NSAPI functions, be sure to see the programmer’s guide for information on making your NSAPI code cacheable as well.

KeepAlive Connections Flushed

A web site that might be able to service 75 requests per second without KeepAlives may be able be able to do 200-300 requests per second when keepalives are enabled. Therefore, as a client requests various items from a

single page, it is important that keepalives are being used effectively. If the `KeepAliveCount` exceeds the `KeepAliveMaxCount`, subsequent KeepAlive connections will be closed (or "flushed") instead of being honored and kept alive.

Checking

Check the `KeepAliveFlushes` and `KeepAliveHits` values. On a site where KeepAlives are running well, the ratio of `KeepAliveFlushes` to `KeepAliveHits` is very low. If the ratio is high (greater than 1:1), your site is probably not utilizing the HTTP KeepAlives as well as it could.

Tuning

To reduce KeepAlive flushes, increase the `MaxKeepAliveConnections` value in the magnus.conf file. The default value is 200. By raising the value, you keep more waiting keepalive connections open.

Warning On Unix/Linux systems, if you increase the `MaxKeepAliveConnections` value too high, the server can run out of open file descriptors. Typically 1024 is the limit for open files on Unix/Linux, so increasing this value above 500 is not recommended.

Log File Modes

Keeping the log files on verbose mode can have a significant affect of performance.

Client-Host, Full-Request, Method, Protocol, Query-String, URI, Referer, User-Agent, Authorization and Auth-User: Because the "obscure" variable cannot be provided by the internal "accelerated" path, the accelerated path will not be used at all. Therefore performance numbers will decrease significantly for requests that would typically benefit from the accelerator, for example static files and images.

iPlanet Web Server has a relaxed logging mode that eases the requirements of the log subsystem. Adding "`relaxed.`*`logname=anything`*" to the "`flex-init`" line in `obj.conf` changes the behavior of the server in the following way: Logging variables other than the "blessed few" does not prevent the accelerated path from being used. If the accelerator is used, the "non-blessed"

variable (which is then not available internally) will be logged as "-". The server does not use the accelerator for dynamic content like CGIs or SHTML, so all the variables would be logged correctly for these requests.

Using Local Variables

The JavaScript virtual machine in iPlanet Web Server implements significant improvements in processing local variables (variables declared inside a function). Therefore, you should minimize the use of global variables (variables declared between the `<server>` and `</server>` tags), and write applications to use functions as much as possible. This can improve the application performance significantly.

Improving Servlet Performance

For information on improving servlet performance, see the *Programmer's Guide to Servlets in iPlanet Web Server*.

Sizing Issues

This section examines subsystems of your server and makes some recommendations for optimal performance:

- Processors
- Memory
- Drive Space
- Networking

Processors

On Solaris and Windows NT, iPlanet Web Server transparently takes advantage of multiple CPUs. In general, the effectiveness of multiple CPUs varies with the operating system and the workload. Dynamic content performance improves the most as more processors are added to the system. Because static content

involves mostly IO and more primary memory means more caching of the content (assuming the server is tuned to take advantage of the memory) more time is spent in IO rather than any busy CPU activity. Our study of dynamic content performance on a four-CPU machine indicate a 40-60% increase for NSAPI and about 50-80% increase for servlets.

Memory

As a baseline, iPlanet Web Server requires 64MB RAM. If you have multiple CPUs, get at least 64MB per CPU. For example, if you have four CPUs, you should install at least 256MB RAM for optimal performance. At high numbers of peak concurrent users, also allow extra RAM for the additional threads. After the first 50 concurrent users, add an extra 512KB per peak concurrent user.

Drive Space

You need to have enough drive space for your OS, document tree, and log files. In most cases 2GB total is sufficient.

Put the OS, swap/paging file, iPlanet Web Server logs, and document tree each on separate hard drives. Thus, if your log files fill up the log drive, your OS will not suffer. Also, you'll be able to tell whether, for example, the OS paging file is causing drive activity.

Your OS vendor may have specific recommendations for how much swap or paging space you should allocate. Based on our testing, iPlanet Web Server performs best with swap space equal to RAM, plus enough to map the document tree.

Networking

For an Internet site, decide how many peak concurrent users you need the server to handle, and multiply that number of users by the average request size on your site. Your average request may include multiple documents. If you're not sure, try using your home page and all its associated subframes and graphics.

Next decide how long the average user will be willing to wait for a document, at peak utilization. Divide by that number of seconds. That's the WAN bandwidth your server needs.

For example, to support a peak of 50 users with an average document size of 24kB, and transferring each document in an average of 5 seconds, we need 240 KBytes/s - or 1920 kbit/s. So our site needs two T1 lines (each 1544 kbit/s). This allows some overhead for growth, too.

Your server's network interface card should support more than the WAN it's connected to. For example, if you have up to 3 T1 lines, you can get by with a 10BaseT interface. Up to a T3 line (45 Mbit/s) you can use 100BaseT. But if you have more than 50 Mbit/s of WAN bandwidth, consider configuring multiple 100BaseT interfaces, or look at Gigabit Ethernet technology.

For an intranet site, your network is unlikely to be a bottleneck. However, you can use the same calculations as above to decide.

4

Using Programs and Objects

- **Extending Your Server With Programs**
- **Working With Configuration Styles**

Chapter

11

Extending Your Server With Programs

This chapter discusses how to install programs on the iPlanet Web Server that dynamically generate HTML pages in response to requests from clients. These programs are known as *server-side applications*. (*Client-side applications*, which are downloaded to the client, run on the client machine.)

This chapter includes the following sections:

- Overview of Server-Side Programs
- Java Servlets and JavaServer Pages (JSP)
- Installing CGI Programs
- Installing Windows NT CGI Programs
- Installing Shell CGI Programs for Windows NT
- Using the Query Handler
- Server-Side JavaScript Programs
- Enabling WAI Services

Overview of Server-Side Programs

Java servlets, JavaScript applications, and CGI programs have different strengths and uses. The following list illustrates the differences between these server-side programs:

- Java servlets are written in Java, which 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. All CGI programs have a standard way of passing information between clients and servers.

Warning Note that you must enable cookies in your browser to run CGI programs.

- JavaScript applications are written in JavaScript, an object-based scripting language. JavaScript is easier to learn than languages such as Java and C and it lends itself to rapid application development.

Note iPlanet Web Server 4.1 does not support server-side Java applets.

Types of Server-Side Applications That Run on the Server

The iPlanet Web Server can run the following types of server-side applications to dynamically generate content:

- Java servlets
- CGI programs
- JavaScript applications

The iPlanet Web Server can also run programs that extend or modify the behavior of the server itself. These programs, known as plug-ins, are written using the Netscape Server Application Programming Interface (NSAPI). For information about writing and installing plug-in programs, see the *NSAPI Programmer's Guide for iPlanet Web Server*.

How Server-Side Applications Are Installed on the Server

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

- For Java servlets, you can configure your server to recognize all files in certain directories as servlets, or you can set up virtual pathnames for servlets, or both. For more information, see "What the Server Needs to Run Servlets and JSPs."
- For CGI programs, you can configure your server to recognize all files with certain filename extensions, or all files in specified directories as CGI programs, or both. For more information, see "Installing CGI Programs," "Installing Windows NT CGI Programs," and "Installing Shell CGI Programs for Windows NT."
- For JavaScript applications, you must check in each application individually through the Application Manager, which you can access from iPlanet Web Server. For more information, see "Installing Server-Side JavaScript Applications."

These installation procedures are described in the following sections.

Java Servlets and JavaServer Pages (JSP)

This section discusses how to install and use Java Servlets and JavaServer Pages on iPlanet Web Server.

This section describes the following topics:

- Overview of Servlets and JavaServer Pages
- What the Server Needs to Run Servlets and JSPs
- Enabling Servlets and JSP
- Making JSPs Available to Clients
- Making Servlets Available to Clients
- Specifying Servlet Directories
- Configuring Global Attributes

- Configuring Servlet Attributes
- Configuring Servlet Virtual Path Translations
- Configuring JRE/JDK Paths
- Configuring JVM Attributes
- Deleting Version Files

Overview of Servlets and JavaServer Pages

iPlanet Web Server supports Java servlets and JavaServer Pages (JSP).

Java servlets are server-side Java programs that can be used to extend the functionality of a web server in much the same way as CGI programs do. Servlets can be thought of as applets that run on the server side without an interface. Servlets are invoked through URL invocation. iPlanet Web Server includes support for JavaSoft's Servlet API at the level of the Java Servlet Development Kit (JSDK) 2.2.1PR.

To develop servlets, use Sun Microsystems' Java Servlet API. For information about using the Java Servlet API, see the documentation provided by Sun Microsystems at:

```
http://www.javasoft.com/products/servlet/2.2/javadoc/
index.html
```

For information about developing servlets for use with iPlanet Web Server, see *Programmer's Guide to Servlets in iPlanet Web Server*.

A JavaServer Page (JSP) is a page, much like an HTML page, that can be viewed in a web browser. However, as well as containing HTML tags, it can include a set of JSP tags that extend the ability of the web page designer to incorporate dynamic content in a page. These tags provide functionality such as displaying property values and using simple conditionals. iPlanet Web Server supports JavaServer Pages (JSP) to the level of JSP API 1.0 compliance.

For information about creating JavaServer Pages, see Sun Microsystem's JavaServer Pages web page at:

```
http://www.javasoft.com/products/jsp/index.html
```

What the Server Needs to Run Servlets and JSPs

iPlanet Web Server includes the Java Runtime Environment (JRE) but not the Java Development Kit (JDK). The server can run servlets using the JRE, but it needs the JDK to run JSP. If you want to run JSP, you must tell iPlanet Web Server to use a custom JDK.

iPlanet Web Server 4.1 requires you to use official versions of JDK, with different platforms requiring different versions, as summarized in Table 11.1.

Table 11.1 Supported JavaSoft JDK Versions by Platform

Platform	JRE/JDK Version
Solaris Sparc	1.2.2_01
Windows NT	1.2.2_01
HPUX	1.2.2_02
AIX	1.2.1
DEC	1.2.1-2
Linux	1.2.2RC3+
IRIX	1.2.1

Check the *iPlanet Web Server Installation and Migration Guide* and the latest *Release Notes* for updates on required JDK versions.

JDK 1.2 (and other JDK versions) are available from Sun Microsystems at:

```
http://www.javasoft.com/products/jdk/1.2/
```

You can specify the path to the JDK in either of the following ways:

- You can specify the path during the server installation process.

  When you install iPlanet Web Server, one of the dialog boxes in the installation process asks if you want to use a custom Java Development Kit (JDK), and if so, you can specify the path to it.

- You can specify it after the server is installed.

To specify the path to the JDK, switch to the Web Server Administration Server, select the **Global Settings** tab, and use the **Configure JRE/JDK Paths** page. You can also change the path to the JDK in this page.

Whether you specify the path to the JDK during installation or later, the path is the folder in which you installed the JDK.

Enabling Servlets and JSP

Before iPlanet Web Server can run servlets, the servlet engine must be enabled. Before the server can serve JSP, the servlet engine must be enabled and JSP must be enabled. (The server cannot serve JSP if servlets are not enabled.)

To enable and disable servlets and JSP in iPlanet Web Server, use the Enable/Disable Servlets/JSP page in the Servlets tab in the Server Manager. This page lets you enable or disable servlets and also enable or disable JSP. If servlets are disabled, you cannot enable JSP.

If servlets are enabled, JSP can be enabled or disabled. However, if you disable servlets, JSP is automatically also disabled. In this case, if you enable servlets later, you must re-enable JSP also if desired.

You can also define a thread pool to be used for servlets. For any server subsystem you can specify which thread pool the servlet's going to run on. For more information about thread pools, see "Adding and Using Thread Pools," on page 167 in Chapter 7, "Configuring Server Preferences."

Making JSPs Available to Clients

You can install JSP files simply by putting them in any directory in or under the document root—you do not need to do anything special to install JSP files—so long as the following conditions are true:

- iPlanet Web Server has been instructed to use the JDK.
- Both servlets and JSP are enabled in the server.

Making Servlets Available to Clients

For servlets, you have a choice of two ways to make a servlet accessible to clients:

- Put the servlet class file in a directory that has been registered with iPlanet Web Server as a servlet directory. For more information, see "Specifying Servlet Directories."
- Define a servlet virtual path translation for the servlet. In this case, the servlet class can be located anywhere in the file system or even reside on a remote machine. For more information, see "Configuring Servlet Virtual Path Translations," on page 282.

You can choose both these options. You can specify a servlet directory and define servlet virtual path translations for servlets outside the servlet directory.

Specifying Servlet Directories

One of the ways to make a servlet accessible to clients is to put it into a directory that has been registered with iPlanet Web Server as a servlet directory. Servlets in registered servlet directories are dynamically loaded when needed. The server monitors the servlet files and automatically reloads them on the fly as they change.

For example, if the `SimpleServlet.class` servlet is in the `servlet` subdirectory of the server's document root directory, you can invoke the servlet by pointing the web browser to:

```
http://your_server/servlet/SimpleServlet
```

You can register any number of servlet directories for iPlanet Web Server. Initially, the web server has a single servlet directory per server instance, which is *server_id*`/docs/servlet/`.

iPlanet Web Server expects all files in a registered servlet directory to be servlets. Any files, including applets, in that directory that have the `.class` extension will be treated as servlets. iPlanet Web Server does not correctly serve other files, such as HTML files or JSPs, that reside in that directory.

The server can have multiple servlet directories, all of which must reside below the primary document directory in the directory hierarchy. You can map servlet directories to virtual directories if desired. For example, you could specify that `http://my_domain.com/products/` invokes servlets in the directory *`server_id`*`/docs/servlet/january/products/servlets/`.

To register servlet directories and to specify their URL prefixes (virtual or not), use the Servlet Directory page in the Servlets tab of the Server Manager. Set the following fields:

URL Prefix. The prefix for accessing the directory. For example, if you want the logical URL `http://`*`servername`*`/plans` to translate to the directory `d:/netscape/server4/docs/plans` then enter `plans` in the URL Prefix field.

Servlet Directory. The absolute pathname to the directory to be registered as a servlet directory, for example, `d:/netscape/server4/docs/plans`. iPlanet Web Server treats all files in the directory as servlets.

Note By default, URLs that are redirected are always escaped. To prevent this, add `escape="no"`. For example:

```
NameTrans fn="redirect" from="/foobar" url-
prefix="index.html" escape="no"
```

Configuring Global Attributes

You can set some global attributes for servlets, including:

- Servlets to run when the server starts up.
- The Session Manager to be used by servlets.
- The Session Manager Args (arguments) to be used by servlets.
- The amount of time the server waits before reloading servlets if they have changed.

To set the global attributes, use the "Configure Global Servlet Attributes" page in the Servlets tab of the Server Manager. Set the following fields:

Startup Servlets. In this field, enter the name of servlets to be loaded when the web server starts up. You do not need to include the `.class` extension.

Session Manager. If you have a session manager class, enter its value here.

Session Manager Args. If you want to specify session manager arguments, enter the values here. Separate multiple *`name=value`* parameters with a comma. Input must be in the format *`name=value,name2=value2`*, and so on. For more information, see Appendix A, "Session Managers," in *Programmer's Guide to Servlets for iPlanet Web Server*.

Reload Interval. This is the interval in seconds the server waits before reloading servlets and JavaServer Pages if they have changed. Specify an integer value here between 0 and 600 inclusive. The default value is 5 seconds.

Configuring Servlet Attributes

If you want to specify input parameters, class paths, or virtual translations for a servlet, you need to individually configure the servlet. Do this in the Configure Servlet Attributes page of the Servlets tag in the Server Manager.

Note When you configure a servlet through the Configure Servlet Attributes page, the servlet is automatically added to the `servlets.properties` file in iPlanet Web Server's `config` directory.

In this page, you can specify the following fields:

Choose Servlet. Specifies the servlet to edit. If no virtual paths have previously been set up, the list is empty. Upon choosing the servlet from this drop-down list, the servlet's information is displayed in the page. (Ignore this field if you are adding a new virtual path entry).

Servlet Name. Specifies an identifier for the servlet. This identifier is used internally by iPlanet Web Server; it is not used in the URL for accessing the servlet. This identifier can be the same name or a different name than the servlet class name.

Servlet Code (class name). Specifies the name of the servlet's main class file. The `.class` extension is optional. Do not specify any directories in this field.

Servlet Classpath. This is the absolute pathname or URL to the directory or zip/jar file containing the servlet. The classpath can point anywhere in the file system. The servlet classpath may contain a directory, a `.jar` or `.zip` file, or a URL to a directory.(You cannot specify a URL as a classpath for a zip or jar file.) If the servlet classpath is not a registered servlet directory, you must

additionally provide a servlet virtual path translation for it (as discussed in "Configuring Servlet Virtual Path Translations" on page 282) to make the servlet accessible to clients.

iPlanet Web Server supports the specification of multiple directories, jars, zips, and URLs in the servlet classpath.

Servlet Args. If the servlet takes additional parameters, enter them here. Separate multiple *name=value* parameters with a comma. Input must be in the format *name=value,name2=value2...* For example, `arg1=45`, `arg2=online`, `arg3="quick shopping"`.

Configuring Servlet Virtual Path Translations

One way to make servlets available to clients is to put them in registered servlet directories. Another way is to define servlet virtual path translations for individual servlets. For example, you could specify that the URL:

```
http://my_domain.com/plans/plan1
```

invokes the servlet defined in

```
server_id/docs/servlets/plans/releaseA/
planP2Version1A.class
```

You can set up servlet virtual path translations for servlets that reside anywhere, on a local or remote file system, or in or out of a registered servlet directory.

Before setting up a servlet virtual path translation, the servlet must have been configured in the Configure Servlet Attributes page of the Servlets tab in the Server Manager, as discussed in "Configuring Servlet Attributes."

To specify a servlet virtual translation path, use the Configure Servlet Virtual Path Translation page in the Servlets tab in the Server Manager. This page has the following fields:

Choose Virtual Path Entry. Specifies a virtual path to modify. If no virtual paths have previously been set up, the list is empty. Upon choosing the virtual path from this drop-down list, the information for the virtual path is displayed in the page. Ignore this field if you are adding a new virtual path entry.

Virtual Path. If you are adding a new path, enter it here. (It's OK to overwrite existing virtual path names.) If you are modifying a path, select the appropriate path from the Choose Virtual Path Entry list to make the path name show up in this field.

The value to enter here is the URL for the virtual path without the `http://servername` part. For example, if you want the virtual path to be `http://servername/virtual/tracker`, enter `/virtual/tracker`.

Servlet Name. Specifies an identifier for the servlet, as entered in the Configure Servlet Attributes page. It's OK if the servlet identifier has not been specified already, but it must be specified before the virtual path will work.

Configuring JRE/JDK Paths

When you install iPlanet Web Server, you can choose to install the Java Runtime Environment (JRE) or you can specify a path to the Java Development Kit (JDK).

The server can run servlets using the JRE, but it needs the JDK to run JSP. The JDK is not bundled with iPlanet Web Server, but you can download it for free from Sun Microsystems at:

```
http://www.javasoft.com/products/jdk/1.2/
```

Regardless of whether you choose to install the JRE or specify a path to the JDK during installation, you can tell the iPlanet Web Server to switch to using either the JRE or JDK at any time. Switch to the Web Server Administration Server, select the **Global Settings** tab, and use the **Configure JRE/JDK Paths** page. You can also change the path to the JDK in this page.

Supply values for the following fields if you select the JDK radio button:

JDK Path. Enter the path for the JDK. This is the directory where you installed the JDK.

JDK Runtime Libpath. Enter the runtime library path for the JDK.

JDK Runtime Classpath. The class path includes the paths to the directories and jar files needed to run the servlet engine, the servlet examples, and any other paths needed by servlets that you add. Values are separated by semicolons. You can add new values to the existing class path, but don't delete the existing value since it includes paths that are essential for servlet operation.

Supply values for the following fields if you select the JRE radio button:

JRE Path. Enter the path for the JRE. This is the directory where you installed the JRE.

JRE Runtime Libpath. Enter the runtime library path for the JRE.

Note If you are not sure of the JDK runtime libpath, the JDK runtime classpath, or the JRE runtime libpath, leave these fields blank to tell the server to use default paths.

Configuring JVM Attributes

You can configure attributes for the Java Virtual Machine (JVM) in the Configure JVM page of the Servlets tab in the Server Manager.

For more information on these options, see *Programmer's Guide to iPlanet Web Server*.

Deleting Version Files

The server uses two directories to cache information for JavaServer Pages (JSP) and servlets:

- `ClassCache`

  When the server serves a JSP page, it creates a `.java` and a `.class` file associated with the JSP and stores them in the JSP class cache, in a directory structure under the `ClassCache` directory.

- `SessionData`

  If the server uses the `MMappedSessionManager` session manager, it stores persistent session information in the `SessionData` directory.

Each cache has a `version` file containing a version number that the server uses to determine the structure of the directories and files in the caches. You can clean out the caches by simply deleting the version file.

When the server starts up, if it does not find the version files, it deletes the directory structure for the corresponding caches and re-creates the version files. Next time the server serves a JSP page, it recreates the JSP class cache. The next time the server serves a JSP page or servlet while using `MMappedSessionManager` session manager, it recreates the session data cache.

If a future upgrade of the server uses a different format for the caches, the server will check the number in the version file and clean up the caches if the version number is not correct.

The Delete Version Files page allows you to delete the files that contain the version number for the JavaServer Pages class cache and the session data cache. This page has the following fields:

Delete the SessionData Version File . Deletes the version file for the session data. When you apply this change, the version file is deleted immediately. The next time the server starts up, it deletes the session data cache and recreates the version file. The next time the server serves a JSP page or servlet while using the `MMappedSessionManager` session manager, it recreates the session data cache.

Delete the ClassCache Version File . Deletes the class cache version file for JSP pages. When you apply this change, the version file is deleted immediately. The next time the server starts up, it deletes the JSP class cache and recreates the version file. The next time the server serves a JSP page, it recreates the class cache.

Installing CGI Programs

This section discusses how to install CGI programs. It has the following sub-sections:

- Overview of CGI
- Specifying a CGI Directory
- Specifying CGI as a File Type
- Downloading Executable Files

In addition, the following sections discuss how to install Windows NT-specific CGI programs:

- Installing Windows NT CGI Programs
- Installing Shell CGI Programs for Windows NT

Overview of CGI

Common Gateway Interface (CGI) programs can be defined with any number of programming languages. On a Unix/Linux machine, you're likely to find CGI programs written as Bourne shell or Perl scripts.

Note Under Unix/Linux, there are extra `CGIStub` processes running that the server uses to aid in CGI execution. These processes are created only during the first access to a CGI. Their number varies depending upon the CGI load on the server. Do not kill these CGIStub processes. They disappear when the server is stopped.

For more information see the discussion regarding `MinCGIStub`, `MaxCGIStub`, and `CGIStubIdleTimeout` in "CGIStub Processes (Unix/Linux)," on page 256 in Chapter 10, "Configuring the Server for Performance."

On a Windows NT computer, you might find CGI programs written in C++ or batch files. For Windows NT, 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 NT CGI programs. See "Installing Windows NT CGI Programs" on page 290 for information about Windows NT CGI.

Note In order to run the command-line utilities, you need to manually set the `Path` variable to include *`server_root`*`/bin/https/bin`.

Regardless of the programming language, all CGI programs accept and return data in the same manner. For information about writing CGI programs, see the following sources of information:

- *Programmer's Guide for iPlanet Web Server*
- *The Common Gateway Interface* at:

  `http://hoohoo.ncsa.uiuc.edu/cgi/overview.html`

- Articles about CGI available on the online documentation web site at:

  `http://www.iplanet.com/docs`

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 all use the file extensions `.cgi`, `.exe`, or `.bat`. The programs can be located in any directory in or under the document root directory.

You can enable both options at the same time if desired.

There are benefits to either implementation. If you want to allow only a specific set of users 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 attempts to interpret any file in that directory as a CGI program. By the same token, if you choose the file type option, your server attempts 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.

Warning Note that you must enable cookies in your browser to run CGI programs.

Note By default, the file extensions for CGI programs are `.cgi`, `.exe` and `.bat.` However, you can change which extensions indicate CGI programs by modifying the MIME types file. You can do this by choosing the Server Preferences tab and clicking the MIME Types link.

Specifying a CGI Directory

To specify a CGI-only directory, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.
2. Click the **CGI Directory** link.

   The CGI Directory window appears.
3. In the URL Prefix field, type the URL prefix 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.

4. 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.
5. Click OK.
6. 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 that any files in those directories will be processed as a CGI file so don't put HTML files in your CGI directory.

Configuring a Unique CGI Directory for Each Software Virtual Server

You can manually configure a unique CGI directory for each software virtual server by manually editing the `obj.conf` file:

```
<Object name="default">
    .
    .
    .
<Client urlhost="www.yourvirtualserver.chm">
 NameTrans fn="pfx2dir" from="/cgi-bin"
 dir="/dir/for/virtual/server/cgi-bin/" name="cgi"
 ...
 </Client>
```

Note that you can not use the Administration Server to accomplish this task.

Specifying CGI as a File Type

To specify CGI programs as a file type, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.
2. Click the **CGI File Type** link.

   The CGI as a File Type window appears.
3. From the Resource Picker, choose the resource you want this change to apply to.
4. Click the **Yes** radio button under Activate CGI as a File Type.
5. Click OK.
6. Save and apply your changes.

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

Downloading Executable Files

If you're using `.exe` as a CGI file type, you cannot 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 the Server Preferences tab and clicking the MIME Types link. 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 NT CGI Programs

This section discusses how to install Windows NT CGI Programs. The following topics are included in this section:

- Overview of Windows NT CGI Programs
- Specifying a Windows NT CGI Directory
- Specifying Windows NT CGI as a File Type

Overview of Windows NT CGI Programs

Windows NT CGI programs are handled much as other CGI programs. You specify a directory that contains only Windows NT CGI programs, or you specify that all Windows NT 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 NT CGI programs, and specify a Windows NT CGI file extension.

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

iPlanet Web Servers support the Windows NT CGI 1.3a informal specification, with the following differences:

- The following keywords have been added to the [CGI] section to support 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 NT 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 multi-part/form-data encoding are not supported.

Specifying a Windows NT CGI Directory

To specify a Windows NT CGI-only directory:

1. From the Server Manager, choose the **Programs** tab.
2. Click the **Win CGI Directory** link.

   The WinCGI Directory window appears.
3. In the **URL Prefix** text field, enter the URL prefix you want to use for this directory.

   That is, the text you type appears as the directory for the Windows NT CGI programs in URLs. For example, if you type `wcgi-programs` as the URL prefix, then all URLs to these Windows NT 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 NT CGI directory you specify in Step 5.

4. Choose whether you want to enable script tracing.

   Click the Yes or No radio button under "Enable Script Tracing?".

   CGI parameters are passed from the server to Windows NT CGI programs through files, which the server normally deletes after the Windows NT 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 NT CGI program brings up is shown when script tracing is enabled.
5. In the **WinCGI Directory** field, enter 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 3.
6. Click OK.
7. Save and apply your changes.

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

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

Specifying Windows NT CGI as a File Type

To specify a file extension for Windows NT CGI files, perform the following steps:

1. From the Server Manager, choose the **Server Preferences** tab.
2. Click the **MIME Types** link.

   The Global MIME Types window appears. For more information on the Global MIME Types, see the "Specifying a Default MIME Type," on page 326 in Chapter 13, "Managing Server Content."
3. Add a new MIME type with the following settings:
   - Type: `type`
   - Content type: `magnus-internal/wincgi`.
   - File Suffix: Enter the file suffixes that you want the server to associate with Windows NT 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. Click the **New Type** button.
5. Save and apply your changes.

Installing Shell CGI Programs for Windows NT

This section discusses how to install Shell CGI Programs for Windows NT. The following topics are included in this section:

- Overview of Shell CGI Programs for Windows NT
- Specifying a Shell CGI Directory (Windows NT)
- Specifying Shell CGI as a File Type (Windows NT)

Overview of Shell CGI Programs for Windows NT

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

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 iPlanet Web Server.

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

Specifying a Shell CGI Directory (Windows NT)

To create a directory for your shell CGI files, perform the following steps:

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 the **Programs** tab.
3. Click the **Shell CGI Directory** link.

   The Shell CGI window appears.
4. In the **URL Prefix** field, enter 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.
5. In the **Shell CGI Directory** field, enter 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.

6. 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 (Windows NT)

You can use the iPlanet Web Server's MIME Types window 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, perform the following steps:

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**.
3. Click the **MIME Types** link.

   The Global MIME Types window appears. For more information on the Global MIME Types, see the "Specifying a Default MIME Type," on page 326 in Chapter 13, "Managing Server Content."
4. Add a new MIME type with these settings:
   - Type: `type`
   - Content type: `magnus-internal/shellcgi`.
   - File Suffix: 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. Click the **New Type** button.
6. Save and apply your changes.

Using the Query Handler

Note The use of Query Handlers is outdated. Although iPlanet Web Server and Netscape Navigator clients still support it, it is rarely used. It is much more common for people to use forms in their HTML pages to submit queries.

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, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.
2. Click the **Query Handler** link.

   The Query Handler window appears.
3. 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.
4. 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.
5. Click OK.
6. Save and apply your changes.

Server-Side JavaScript Programs

To allow iPlanet Web Server to run Server-Side JavaScript programs, you need to enable Server-Side JavaScript, which you can do in the Activate Server Side JavaScript page in the Programs tab of the Server Manager.

To install and manage Server-Side JavaScript programs on the server, use the JavaScript Application Manager.

The following topics are included in this section:

- Activating Server-Side JavaScript
- Running the Application Manager
- Securing the Application Manager
- Installing Server-Side JavaScript Applications
- Application URLs
- Controlling Access to a Server-Side JavaScript Application
- Modifying Installation Parameters
- Removing a Server-Side JavaScript Application
- Starting, Stopping, and Restarting a Server-Side JavaScript Application
- Running a Server-Side JavaScript Application
- Configuring Default Settings

For more information about writing JavaScript applications, see *Writing Server-Side JavaScript Applications* at:

```
http://www.iplanet.com/docs
```

Activating Server-Side JavaScript

To enable Server-Side JavaScript on iPlanet Web Server, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.
2. Click the **Server Side Javascript** link.

   The Activate Server Side Javascript window appears.

3. Under **Activate the Server Side Javascript application environment**, click the **Yes** radio button.

4. If you want to require the administration server username and password before allowing access to the Application Manager, select the second Yes radio button.

   For more information on securing the application manager, see "Securing the Application Manager" on page 301.

5. Click OK.

6. Save and apply your changes.

Running the Application Manager

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.

To run the Application Manager, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.

2. Click the **Server Side Javascript** link.

   The Activate Server Side Javascript window appears.

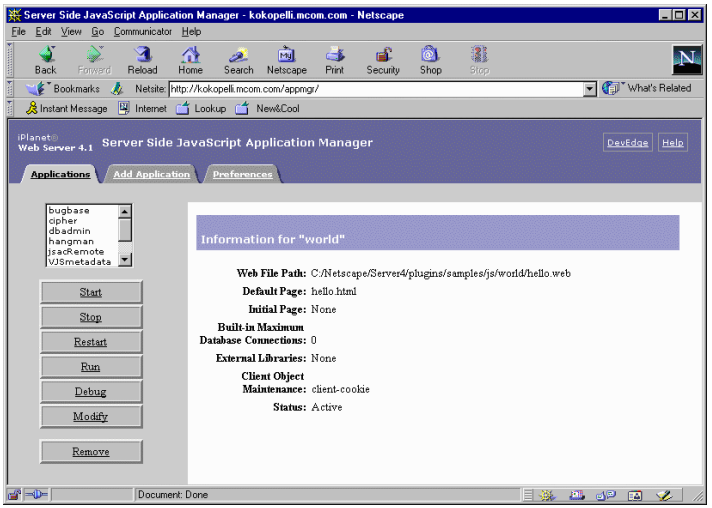
3. Click the link to the Application Manager.

4. Click OK.

   You can also run the Application Manager by loading the following URL in Navigator: `http://server.domain/appmgr`.

   iPlanet Web Server displays the following page:

Figure 11.1 The Application Manager



   The Application Manager displays all applications currently installed on the server in a scrolling list in the left frame.

5. Select an application by clicking its name in the scrolling list.

   For the selected application, the right frame displays the following information:

   - The application name at the top of the frame.
   - The path of the application `.web` file on the server. (The `.web` file is the compiled JavaScript application.)
   - 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.

6. 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.

7. Click the **Add Application** tab at the top to add a new JavaScript application.

8. Click the **Preferences** tab at the top to configure the default settings to use when adding a new application.

Securing the Application Manager

The Application Manager runs on iPlanet Web Server. It is installed into the `js/appmgr` directory. It can be accessed with the URL:

`http://`*`server`*`.`*`domain`*`:port/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 username and password for access to the Application Manager, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.

2. Click the Server Side Javascript link.

   The Activate Server Side Javascript window appears.

3. Under "Require administration server password for Server Side Javascript Application Manager" click the Yes radio button.

4. Click OK.

5. Save and apply your changes.

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 Application 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 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, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.
2. Click the **Server Side Javascript** link.

   The Activate Server Side Javascript window appears.
3. Click the link to the Application Manager.
4. Click the **Add Application** tab at the top of the page.

   The Add Application window appears.
5. 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:port/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 305.

6. In the **Web File Path** field, type the absolute path to the `.web` file for the application.

   This is a required field.

7. 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.

8. In the **Initial Page** 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.

   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. (This is provided for backward compatibility with applications that use database objects; for new applications that use a `dbpool`, ignore this field. See Chapter 8, "Connecting to a Database," in the *Server-Side JavaScript Guide* for how to set this parameter for a `dbpool`.)

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. For additional information on client objects, refer to *Writing Server-Side JavaScript Applications* at:

    `http://www.iplanet.com/docs`

    The choices for the client object maintenance technique are:

- **client-cookie**—Specifies that the JavaScript runtime engine on the server should use the Netscape cookie protocol to transfer the properties of the client object and their values to the client. It creates one cookie per client property. The properties are sent to the client once, in the response header of the generated HTML page.
- **client-url**—Specifies that the runtime engine on the server should transmit the properties of the client object and their values to the client by appending them to each URL in the generated HTML page. Consequently, the properties and their values are sent as many times as there are links on the generated HTML page, resulting in the largest increase in network traffic of all of the maintenance techniques.
- **server-ip**—Specifies that the server should index the data structure based on the application and the client's IP address. This technique is the fastest, because it does not require sending any information to the client. Since the index is based on both the application and the IP address, this technique creates a separate index for every application/client pair running on the server.
- **server-cookie**—Specifies that the server should use a long unique name, generated by the runtime engine, to index the data structure on the server. The runtime engine uses the Netscape cookie protocol to store the generated name as a cookie on the client. It does not store the property names and values as cookies. For this reason, this technique creates a single cookie, whereas the client-cookie technique creates a separate cookie for each property of the client object.
- **server-url**—Specifies that the server should use a long unique name, generated by the runtime engine, to index the data structure on the server. In this case, rather than making that generated name be a cookie on the client, the server appends the name to each URL in the generated HTML page. Consequently, the name is sent as many times as there are links on the generated HTML page. (Property names and values are not appended to URLs, just the generated name.)

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 for resources in 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:port/appName/page.html
```

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

You can also access the application with the URL

```
http://server.domain:port/appName/
```

since the server knows the default page to open.

For example, if your server is named myserver and your domain name is mozilla.com, the application name is world, and the default page is hello.html, you can access the application with either of the following URLs:

```
http://myserver.mozilla.com/world/hello.html
```

or

```
http://myserver.mozilla.com/world/
```

When a client requests an application URL, the server runs the Server-Side JavaScript code inside the default page then sends the resultant HTML page 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 Chapter 12, "Working With Configuration Styles." For more information on restricting access to part of your server, see Chapter 14, "Controlling Access to Your Server."

Modifying Installation Parameters

To modify an application's installation parameters, open the Application Manager as described in the section "Running the Application Manager," on page 299. Then 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, open the Application Manager as described in "Running the Application Manager," on page 299, then 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, open the Application Manager as described in "Running the Application Manager," on page 299, and then click Start. If the application starts successfully, clients can invoke the application.

To stop an active application, click Stop. The application's status changes to "stopped," and clients can no longer invoke 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. Before any changes 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:port/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:

- Open the Application Manager as described in "Running the Application Manager," on page 299, then select the application name in the Application Manager, and click Run. A new Navigator window accesses the application.
- Enter 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

To configure default settings for new applications, open the Application Manager as described in "Running the Application Manager," on page 299, and then click the Preferences tab. 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.

Enabling WAI Services

Note Web Application Interface (WAI) is provided in iPlanet Web Server 4.x, but is not guaranteed to be supported in future releases. We recommend that you do not develop new WAI applications.

WAI services are a kind of plug-in that uses the **Common Object Request Broker Architecture (CORBA)**. WAI applications can be written in C, C++, or Java, and they interact with iPlanet Web Server over Internet Inter-ORB Protocol (IIOP). A WAI application runs within its own process. WAI applications need an object request broker (ORB) to work.

iPlanet Web Server 4.x does not ship with an object request broker (ORB). Before using WAI applications, you must install Visibroker 3.3+ from Inprise. You can get Visibroker3.3+ from the Inprise web site at:

```
http://www.inprise.com/products/
```

After installing Visibroker 3.3+, you will need to install WAI on your iPlanet Web Server. You can do this by running through the installation process, and choosing to install only WAI.

After you have installed WAI, the next step is to enable it on your server. Enabling WAI services essentially turns on Internet Inter-ORB Protocol (IIOP) support in the server.

To enable WAI services on your server, perform the following steps:

1. From the Server Manager, choose the **Programs** tab.
2. Click the **WAI Handler** link.

   The WAI Administration window appears.
3. To enable WAI services, click the **Yes** radio button.
4. Save and apply your changes.

For more information about WAI, see *Writing Web Applications with WAI* at:

```
http://www.iplanet.com/docs
```

Chapter

12

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.

This chapter includes the following sections:

- Creating a Configuration Style
- Removing a Configuration Style
- Editing a Configuration Style
- Assigning a Configuration Style
- Listing Configuration Style Assignments

Creating a Configuration Style

To create a configuration style, perform the following steps:

1. Access the Administration Server and click the **Servers** tab.
2. In the Manage Servers area, select the desired server and click **Manage**.

   iPlanet Web Server displays the Server Manager Preferences page, as shown in Figure 1.1. on page 47 of Chapter 1, "Introduction to iPlanet Web Server."
3. Choose the **Styles** tab.
4. Click the **New Style** link.
5. Type the name you want to give the configuration style. Click OK.

   iPlanet Web Server displays the Edit a Style page.
6. From the drop-down list, choose a configuration style to edit and click **Edit this Style**.
7. From the list of links available, click the category you want to configure for your style.

   You can configure the information listed in Table 12.1.
8. Fill out the form that appears, and click OK.
9. Repeat step 4 and step 5 to make any other configuration changes to the configuration style. Click OK.
10. 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.

Table 12.1 Configuration Style Categories

Category	Description
CGI file type	Allows you to activate CGI as a file type. For more information about CGIs, see "Installing CGI Programs," on page 285 in Chapter 11, "Extending Your Server With Programs."
Character Set	Allows you to change the character set for a resource. For more information about character sets, see "Changing the Character Set," on page 333 in Chapter 13, "Managing Server Content."
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 297 in Chapter 11, "Extending Your Server With Programs."
Document Footer	Allows you to add a document footer to a server resource.
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 173 in Chapter 7, "Configuring Server Preferences."
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 172 in Chapter 7, "Configuring Server Preferences."
Log preferences	Allows you to set preferences for access logs. For more information about log preferences, see "Setting Log Preferences," on page 191 in Chapter 8, "Understanding Log Files."
Restrict Access	Allows you to restrict access to the entire server or parts of it. For more information about access control, see Chapter 14, "Controlling Access to Your Server."

Table 12.1 Configuration Style Categories

Category	Description
Server Parsed HTML	Allows you to specify whether the server parses files before they are sent to the client.
Symbolic links (Unix/Linux)	Allows you to limit the use of file system links in your server. For more information about symbolic links, see "Restricting Symbolic Links (Unix/Linux)," on page 181 in Chapter 7, "Configuring Server Preferences."

For more information, see "The Create a New Style Page," in the online help.

Removing a Configuration Style

Before removing a configuration style, remove assignments 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, perform the following steps:

1. Access the Server Manager and choose the **Styles** tab.
2. Click **List Assignments** link.
3. Select **Edit Style Assignment** you want to remove.
4. Click **Remove this Assignment**.
5. Click the **Remove Style** link.
6. Select the configuration style you want to remove and click OK.

For more information, see "The Remove a Style Page," in the online help.

Editing a Configuration Style

To edit a configuration style, perform the following steps:

1. Access the Server Manager and choose the **Styles** tab.
2. Click the **Edit Style** link.
3. Select the configuration style you want to edit and click the **Edit this style** button.
4. From the list of links available, click the category you want to configure for your style.

   For more information on these categories, see the section "Creating a Configuration Style" on page 312.
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. Click OK.
7. 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.

For more information, see "The Edit a Style Page," in the online help.

Assigning a Configuration Style

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

To assign a configuration style, perform the following steps:

1. Access the Server Manager and choose the **Styles** tab.
2. Click the **Assign Style** link.
3. 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.
4. Select the configuration style you want to apply. To remove any configuration style previously applied to the resource, apply the None configuration style. Click OK.

For more information, see "The Apply a Configuration Style Page," in the online help.

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 the configuration style assignments, perform the following steps:

1. Access the Server Manager and choose the **Styles** tab.
2. Click the **List Assignments** link.

   iPlanet Web Server displays the List Assignments page, showing the configuration styles you applied to server resources.
3. To edit a configuration style assignment, click the Edit link next to the configuration style name.

For more information, see "The View, Edit, or Remove Style Assignments Page," in the online help.

5

Managing Content and Access

- **Managing Server Content**
- **Controlling Access to Your Server**
- **Configuring Web Publishing**
- **Using Search**

Chapter

13

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.

This chapter contains the following sections:

- Changing the Primary Document Directory
- Setting Additional Document Directories
- Customizing User Public Information Directories (Unix/Linux)
- Enabling Remote File Manipulation
- Configuring Document Preferences
- Setting Up Hardware Virtual Servers
- Setting Up Hardware Virtual Servers for ISPs
- Setting up Software Virtual Servers
- Changing the Character Set

Changing the Primary Document Directory

The primary document directory or document root is the central directory where you store all the files you want to make available to remote clients. You specified a primary document directory when you installed the iPlanet Web Server software. This section describes how to change the primary document directory from what you specified in the installation process.

The primary document directory provides an easy way to restrict access to the files on your server. It also makes it easy to 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\server4\docs`, a request such as `http://www.mozilla.com/products/info.html` tells the server to look for the file in `C:\Netscape\Server4\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, use the Primary Document Directory page in the Server Manager. For more information, see the online help.

Note Each server instance should have its own primary document directory. If server instances share primary document directories, users could simultaneously modify a document without knowing it.

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 directory 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.

Unix/Linux: If you expect web publishing users to publish documents to a directory, you need to set the Unix/Linux file permissions to give them write access to that directory. You should also disable write permissions for directories you do not want them to publish to.

To add additional document directories, use the Additional Document Directories page in the Server Manager.

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

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

You should restrict access to these directories so that users cannot write to them. A sample ACL for the `/publisher` directory would be:

```
deny (all) anyone;

allow (rxli) all;

allow (wd) privileged_user;
```

Customizing User Public Information Directories (Unix/Linux)

Sometimes users want to maintain their own web pages. You can configure public information directories that let all the users on your server create home pages and other documents without your intervention.

Another way to do this is to create a URL mapping to a central directory that all of your users can modify.

With this system, clients can access your server with a certain URL that the server recognizes as a public information directory. For example, suppose you choose the prefix ~ and the directory `public_html`. If a request comes in for `http://www.iplanet.com/~jdoe/aboutjane.html`, the server recognizes that `~jdoe` refers to a users' public information directory. It looks up `jdoe` in the system's user database and finds Jane's home directory. The server then looks at `~/jdoe/public_html/aboutjane.html`.

To configure your server to use public directories, you need to choose a user URL prefix. The usual prefix is ~ because the tilde character is the standard Unix/Linux prefix for accessing a user's home directory. Next, you need to choose the subdirectory in the user's home directory where the server looks for HTML files. A typical directory is `public_html`.

The server needs to know where to look for a file that lists users on your system. The server uses this file to determine valid usernames and to find their home directories. If you use the system password file for this purpose, the server uses standard library calls to look up users. Alternatively, you can create another user file to look up users. You can specify that user file with an absolute path.

Each line in the file should have this structure (the elements in the `/etc/passwd` file that aren't needed are indicated with `*`):

```
username:*:*:groupid:*:homedir:*
```

Restricting Content Publication

In some situations a system administrator may want to restrict what user accounts are able to publish content via User Document Directories. This can easily be accomplished by adding a trailing slash to the user's home directory path in the `/etc/passwd file`:

```
jdoe::1234:1234:John Doe:/home/jdoe:/bin/sh
```

becomes:

```
jdoe::1234:1234:John Doe:/home/jdoe/:/bin/sh
```

When this modification is made, iPlanet Web Server will not serve pages from this user's directory. The browser requesting the URI receives a "404 File Not Found" error and a 404 error will be logged to the web server access log. No error will be logged to the errors log.

If, at a later time, the system administrator decides to allow this user to publish content the trailing slash should be removed from the `/etc/passwd` entry followed by restarting the web server.

Loading the Entire Password File on Startup

You also have the option of loading the entire password file on startup. If you choose this option, the server loads the password file into memory when it starts, making user lookups much faster. If you have a very large password file, however, this option can use too much memory.

Using Configuration Styles

Finally, you can apply a configuration style for the server to control access to directories from public information directories. This prevents users from creating symbolic links to information you do not want made public.

To set up user directories, use the User Document Directories page in the Server Manager.

Enabling Remote File Manipulation

When you enable remote file manipulation, clients are able to upload files, delete files, create directories, remove directories, list the contents of a directory, and rename files on your server. The file `obj.conf` in the directory *`server_root`*`/https-`*`serve-id`*`/config` contains the commands that are activated when you enable remote file manipulation. By activating these commands, you allow remote browsers to change your server's documents. You should use access control to restrict write access to these resources to prevent unauthorized tampering.

Note When you enable remote file manipulation, you need to disable Web Publishing functions on your server. When you use Web Publishing, you need to disable remote file manipulation. The two sets of functions cannot operate simultaneously. You can use both remote file manipulation and web publishing functions by manually setting the function to be called for each individual remote method invocation. However, proceed with caution as mixing the functions can affect the server's web publishing state.

Unix/Linux: You must have the correct permissions for your files or this function will not work; that is, the document root user must be the same as the server user.

To enable remote file manipulation, use the File Manipulation page in the Server Manager.

Configuring Document Preferences

You use the Document Preferences page to set document preferences. This section discusses these topics:

- Entering an Index Filename
- Selecting Directory Indexing
- Specifying a Server Home Page
- Specifying a Default MIME Type
- Parsing the Accept Language Header

Entering an Index Filename

If a document name is not specified in the URL the server automatically displays the index file. The default index files are `index.html` and `home.html`. If more than one index file 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` and `home.html`, the server looks for `index.html` and if it doesn't find it looks for `home.html`.

To enter an index filename, edit the Index Filenames field in the Document Preferences page of the Server Manager.

Selecting 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 by searching 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 the previous section, "Entering an Index Filename" on page 324). 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.

To select directory indexing, use the Document Preferences page in the Server Manager.

Warning If your server is outside the firewall, turn off directory indexing as well as web publishing (from the Web Publishing State page) to ensure that your directory structure, filenames, and web publishing features are not accessible.

Specifying a 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 in the Document Preferences page and uses that for the home page. However, you can also specify a file to use as the home page in the Document Preferences page of the Server Manager. For more information, see the online help.

Specifying a 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 the Global MIME Types page in the online help.

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`

To specify a default MIME type, use the Document Preferences page of the Server Manager. For more information, see the online help

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 the section "Using the Accept Language Header" on page 475.

To parse the accept language header, use the Document Preferences page in the Server Manager.

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. iPlanet Web Server can respond to up to 256 IP addresses.

Note If you are using more than 100 hardware virtual servers, you should use the method described in the section "Setting Up Hardware Virtual Servers for ISPs" on page 328 for setting up hardware virtual servers.

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. Alternatively, you can also configure multiple hardware virtual servers on the same IP address by assigning different port numbers for each hardware virtual server. For more information, see "Adding a Server: Running Multiple Servers" on page 58.

Unix/Linux. Before you set up hardware virtual servers, make sure you specified a specific bind-to-address for your server in the Network Settings page (from the iPlanet Web Server, choose Server Preferences and then click Network Settings). If you left the Bind To Address field blank, you may experience errors when using hardware virtual servers. If you are an ISP using hardware virtual servers, the bind-to-address should be your main IP address.

You can set up hardware virtual servers through the Hardware Virtual Servers page in the Server Manager. For more information, see the online help.

Setting Up Hardware Virtual Servers for ISPs

ISPs that need to support more than 256 IP addresses or that want the server to use less memory can use the ISP-version of the hardware virtual server function. As with default hardware virtual servers (discussed in the previous section), ISP-version hardware virtual servers allow you to configure your server to respond to multiple IP addresses without installing multiple servers, but you can configure your server to support an arbitrary number of IP addresses.

ISP-version 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 "Changing Network Settings" on page 66.

For HP servers, the number of virtual servers must work well with the `max_thread_proc` entry in the HP-UX kernel and `RqThrottle`. Since threads are never "released," but moved to another pool, the number of threads used can get quite high when using hardware virtual servers in the object model. For more information, see "About RqThrottle (Maximum Simultaneous Connections)" on page 258.

Note If you set up this hardware virtual server function, make sure that the B**ind to Address** field in the Network Settings page is blank (choose **Server Preferences** and click **Network Settings**).

This section includes the following topics:

- To Set Up Hardware Virtual Servers For an ISP
- To Edit a Server Instance
- To Remove a Server Instance
- Migrating Hardware Virtual Server Configuration Files

To Set Up Hardware Virtual Servers For an ISP

To set up hardware virtual servers for an ISP, perform the following steps:

1. Uncomment the line for setting up hardware virtual servers for an ISP in the `index.1st` file in *`server_root`*`/bin/https/httpadmin/html` directory.

   The default `index.lst` file comments out the line for the ISP-version hardware virtual server. You must uncomment the line containing "`Option:perl/virtual, Hardware Virtual Servers`" and comment out the line containing "`Option:multiple, Hardware Virtual Servers`".

2. From the Server Manager, choose the **Content Management** tab.

   Click **Content Management** even if it's already selected to make sure the file name change is picked up by the server.

3. Click **Hardware Virtual Servers**.

   The Hardware Virtual Servers page appears.

4. Enter the server's IP address in the IP field.

5. Enter the primary document directory in the Doc Root field, and click OK.

   You must type in the absolute path, such as C:`/Netscape/server4/docs`.

6. Click **Apply** in the top right portion of the Server Manager to apply your changes.

To Edit a Server Instance

To edit a server instance, perform the following steps:

1. Click **Edit** on the line for the server instance you want to edit.
2. On the Hardware Virtual Servers page, enter the new IP address and document root, and click OK.
3. Click **Apply** in the top right portion of the Server Manager to apply your changes.

To Remove a Server Instance

To remove a server instance, perform the following steps:

1. Click **Remove** on the line for the server instance you want to remove.
2. Click OK in the confirmation dialog box.
3. Click **Apply** in the top right portion of the Server Manager to apply your changes.

The ISP-hardware virtual servers are listed in the `virtual.conf` configuration file. This file lists the IP addresses you entered through the Server Manager and the document root to which they apply.

You can return to using the default hardware virtual server function, by performing the following steps:

1. From the Server Manager, choose **Content Management** tab.
2. Click **Hardware Virtual Servers**.

   The Hardware Virtual Servers Page appears.
3. Click **No** to deactivate the ISP-version hardware virtual server function, then click OK.
4. Click **Save and Apply**.

Migrating Hardware Virtual Server Configuration Files

If you run multiple IP addresses using the `obj.conf` file, you may have restrictions using virtual servers. For better reliability, you can migrate from the `obj.conf` file to the `virtual.conf` file by running the following script:

```
server_root/bin/https/httpadmin/bin/vserverupgrd -r
server_root -p administration_port -i https-server-id
```

When you do this, the executable removes all the `NameTrans` directives, for individual hardware virtual servers, from the obj.conf file that corresponds to the `config` directory *`server_root`*`/http-`*`server-id`*, and replaces them with corresponding directives in a virtual.conf file, located in the same directory. It also references the virtual.conf file from the `magnus.conf`. file in the *`server_root`*`/https-`*`server-id/`*`config/` directory, and removes addresses that are found in the `magnus.conf` file.

To use the Server Manager for the hardware virtual servers that are uses with ISPs, see the section "Setting Up Hardware Virtual Servers for ISPs" on page 328.

Setting up Software Virtual Servers

Setting up a software virtual server enables you 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.siroe.com` and `www.iplanet.com` resolve to 192.3.4.5, then set up software virtual servers to handle both server names (for example, `http://www.siroe.com/` and `http://www.iplanet.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.siroe.com/` and `http://www.iplanet.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.

Note You cannot use Netscape Navigator version 1.x with software virtual servers. You should use Netscape Navigator 3.01 or later.

You can set up a software virtual server using the Software Virtual Servers page in the Content Management tab of the Server Manager.

For information on how to configure a unique CGI directory for specific software virtual servers, see "Configuring a Unique CGI Directory for Each Software Virtual Server," on page 289 in Chapter 11, "Extending Your Server With Programs."

Adding a Doc Root for Software Virtual Servers

You can assign a unique doc root for each software virtual server.

Note You can not use the Administration Server to accomplish this task.

To assign every software virtual server its own doc root, add the following code to the `obj.conf` file:

```
NameTrans fn="document-root" root="/any/other/
directory/root"
```

Changing the Character Set

The character set of a document is determined in part by the language it is written in. You can override a client'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. Examples are:

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

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

- `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 i`so_8859-1`:

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

To change the character set, use the International Characters page in the Server Manager.

Chapter

14

Controlling Access to Your Server

This chapter discusses the various methods you can use to control access to the Administration Server and to the files or directories on your web site. For example, for the Administration Server, you can specify who has full control of all the servers installed on a machine and who has partial control of one or more servers. Before you can use access control on the Administration Server, you must enable distributed administration from the Distributed Administration page and set up an administration group in your LDAP database. This chapter assumes you have already configured distributed administration and have defined users and groups in your LDAP database.

You should also ensure the security of the web server as discussed in Chapter 5, "Working with Server Security."

This chapter contains the following sections:

- What Is Access Control?
- How Access Control Works
- Restricting Access to Your Web Site
- Access Control Examples
- Access Control For Web Publishing

What Is Access Control?

Access control allows you determine who can access iPlanet Web Administration Server and which servers and tabs (also called programs) they can access as well as who can access the files or directories on your web site.

You can control access to the entire server or to parts of the server such as specific tabs or pages in the Administration Server or the files or directories on your web site. 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.

You can use two methods for controlling access:

- **User-Group.** This method requires users to enter a username and password before accessing the server. The server compares the information in a client certificate or the client certificate itself with a directory server entry. This methods requires the use of a directory server. If you choose to use client certificates, you should increase the value of the `AcceptTimeout` directive in `magnus.conf`.

- **Host-IP.** This method requires the user to access the web server from a specific computer, where the web server recognizes the computer by either its hostname or its IP address. This methods does not require a directory server.

This section includes the following topics:

- Setting ACL User Cache Time
- User-Group Authentication
- Host-IP Authentication
- Access Control Files

Setting ACL User Cache Time

To control the amount of time that ACL user cache is valid, use the `ACLCacheLifetime` directive in the `magnus.conf` file. Each time an entry in the cache is referenced, its age is calculated and checked against `ACLCacheLifetime`. The entry is not used if its age is greater than or equal to the `ACLCacheLifetime`. The default value is 120 seconds. If this value is set to 0, the cache is turned off. If you use a large number for this value, you may need to restart iPlanet Web Server when you make changes to the LDAP entries. For example, if this value is set to 120 seconds, iPlanet Web Server might be out of sync with the LDAP server for as long as two minutes. If your LDAP is not likely to change often, use a large number.

The maximum number of entries that can be held in the cache is configurable as of iPlanet Web Server 4.0, using the `magnus.conf` parameter, `ACLUserCacheSize`. The default value for this parameter is 200, which is the fixed size of the cache in ES 3.x. New entries are added to the head of a list, and entries at the end of this list are recycled to make new entries when the cache reaches its maximum size.

The maximum number of group memberships that can be cached per user entry is configurable as of ES 4.x, using the `magnus.conf` parameter, `ACLGroupCacheSize`. The default value for this parameter is 4, although in ES 3.x it has a fixed value of 1. Unfortunately non-membership of a user in a group is not cached, and will result in several Directory Server operations for each such check on every request.

User-Group Authentication

User-Group authentication requires users to authenticate themselves before getting access to the Administration Server or the files or directories on 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 Communicator. The first method of getting the username and password is the basic method, which can be done with or without encryption. The latter method of using client certificates is the SSL method, which must be done with encryption on. For information on using SSL, see Chapter 5, “Working with Server Security.”

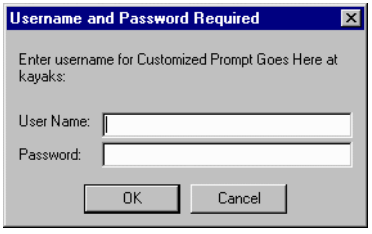
Username and Password Authentication

To require users to enter a username and password to get access to the web server or your web site, you must store the list of users and groups in an LDAP database such as the Netscape Directory Server. The directory server can be running on the same machine as the web server, or you can use a directory server installed on a remote machine.

When users attempt to access a resource that has User-Group authentication in the Administration Server or on your web site, the web browser displays a dialog box asking the user to enter a username and password. The server receives this information encrypted or unencrypted, depending on whether encryption is turned on for your server.

After entering the username and password, the user either sees the Server Administration page if logging in to iPlanet Web Administration Server, the file or directory listing requested if logging in to a web site, or a message denying access if the username or password was invalid. You can customize the access denied message that unauthorized users see through the Access Denied Response page. Figure 14.1 shows the authentication dialog box. This dialog box displays a customized login prompt message.

Figure 14.1 Users see this dialog box when authenticating themselves to the server.



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

The server maintains two connections to the directory server. One of these is used to authenticate users, by doing an LDAP bind as the specified user. The other is permanently bound as the `binddn` specified in the Configure Directory Service page, and is used for locating user entries and checking group

memberships. Only one HTTP request thread can access the directory server at a time, which means that a global lock controls access to both LDAP connections. This can be a potential performance bottleneck, especially when combined with the fixed size of the ACL user/group cache.

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 if certificates are published in an LDAP directory.

When a server with client authentication enabled receives a request, the server performs the following actions:

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, and the authorization fails.
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 "Using the certmap.conf File" on page 136 for more information on setting up the certificate mapping file.
3. If the certificate maps correctly, then the web server checks the ACL rule specified for that user. Therefore, even though the certificate maps correctly, if the ACL denies the user access, the rule can deny the request.

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 with access control, choose the SSL authentication methods you want to use from the Encryption Preferences page (in the Preferences tab, click Encryption Preferences). To require client authentication for the entire server, select "Require Client Certificates (regardless of access control)" in the Encryption Preferences page.

Note Only the SSL authentication method requires modification to the `certmap.conf` file. Allowing client authentication for all connections to the server does not.

In order for a client to successfully gain access to a SSL authenticated resource requiring client certificates, the client must install a certificate on their browser which is from a certificate authority trusted by the web server. It may be necessary to have the same client certificate published in a directory server if the web server's `certmap.conf` file is configured to compare the entire certificate between the client's certificate in the browser and the client certificate in the directory server entry. However, the `certmap.conf` file can be configured so that it only compares selected information from the certificate to the entry in the directory server. For example, you can configure the `certmap.conf` file so that the server only compares a user ID and an email address in the browser certificate with the directory server entry. In such a case, it would not be necessary to publish the entire client certificate to the directory server since only the user ID and email address must match to gain access.

Host-IP Authentication

You can limit access to the Administration Server or the files or directories on your web site by making them available only to clients 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. 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 machine does not have access, the user will see a message denying access. For information on customizing this message, see "Responding When Access is Denied" on page 357.

Note It is possible for more than one person to have access to a particular system. For this reason, Host-IP authentication is more 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 on a particular computer before getting access.

IP authentication does not require DNS to be configured on your server. If you want to use hostname authentication, however, you must have DNS running in your network and your server must be configured to use it. You can enable DNS on your server through the Performance Tuning page in the Preferences tab.

Enabling DNS degrades the performance of iPlanet Web Server since the server is forced to do DNS look-ups. To reduce the effects of DNS look-ups on your server's performance, resolve IP addresses only for access control and CGI instead of resolving the IP address for every request. To do this, add the line "`iponly=1`" to the line that begins: `AddLog fn="flex-log" name="access"` in your `obj.conf` file. The resulting line is as follows:

```
AddLog fn="flex-log" name="access" iponly=1
```

Access Control Files

When you use access control on the Administration Server or the files or directories on your web site, the settings are stored in a file with the extension `.acl`. Access control files are stored in the directory *`server_install`*`/httpacl` where *`server_install`* is the location where the server is installed. For example, if you installed the server in /`usr/netscape/server4`, the ACL files for both the Administration Server and each server instance configured on your server would be located in `/usr/netscape/server4/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 iPlanet Web Server to restrict access, you'll have these two files. However, if you want 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 such as restricting access to the server based on the time of day or day of the week.

Note When server users change ACLs through Web Publisher, the ACL configuration files are modified, and you receive a JavaScript notification alerting you of the change.

Also, you can manually create and edit `.acl` files to customize access control. For example, if you want to use an Oracle or Informix database of users instead of an LDAP database, you need to use the access control API to program a

hook into the server's access control structure. This API is written in the C programming language. For more information on the API, see the iPlanet documentation site at `http://www.iplanet.com/docs.`

For more information on access control files and their syntax, see "ACL File Syntax" on page 466.

How Access Control Works

When the server gets a request for a page, the server uses the rules in the ACL file to determine if it should grant access or not. The rules can reference the hostname or IP address of the computer sending the request. The rules can also reference users and groups stored in the LDAP directory.

For example, the following ACL file contains the two default entries for the Administration Server (`admin-serv`) plus an additional entry that allows users in the "admin-reduced" group to access the Preferences tab in the Administration Server.

```
version 3.0;

# The following "es-internal" rules protect files such
# as icons and images related to iPlanet Web Server.
# These "es-internal" rules should not be modified.

  acl "es-internal";
  allow (read, list, execute,info) user = "anyone";
  deny (write, delete) user = "anyone";

# The following "default" rules apply to the entire document
# directory of iPlanet Web Server. In this example, the rules
# are set up so that "all" users in the directory server are
# allowed to read, execute, list, and get information.
# The "all" users are not allowed to write to or delete any files.
# All clients that accesses the document directory of the web
# server will be required to submit a username and password
# since this example is using the "basic" method of
# authentication. A client must be in the directory server
# to gain access to this default directory since "anyone"
# not in the directory server is denied, and "all" in the
# directory server are allowed.

  acl "default";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
```

```
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (user = "all");

# The following rules deny access to the directory "web"
# to everyone not in the directory server and deny everyone
# in the directory server who is not in GroupB.
# Only the users in GroupB are allowed read, execute, list,
# and info permissions. GroupA can not gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory “my_stuff”. Furthermore, members
# of GroupB can not write or delete files.

  acl "path=/export/user/990628.1/docs/my_stuff/web/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");


  allow (read,execute,list,info)
  (group = "GroupB");


# The following rule denies everyone not in the directory
# server and denies everyone in the directory server except
# user with the ID of "SpecificMemberOfGroupB". The ACL rule
# in this setting also has a requirement that the user
# connect from a specific IP address. The IP address setting
# in the rule is optional; it has been added to for extra
# security. Also, this ACl rule has a Customized prompt
# of "Presentation Owner". This Customized prompt appears
# in the username and password dialog box in the client's
# browser.


  acl "path=/export/user/990628.1/docs/my_stuff/web/presentation.html";
  authenticate (user,group) {
     database = "default";
     method = "basic";
     prompt = "Presentation Owner";
  };
  deny (all)
  (user = "anyone" or group = "my_group");
  allow (all)
  (user = "SpecificMemberOfGroupB") and
  (ip = "208.12.54.76");
```

```
# The following ACL rule denies everyone not in the directory
# server and everyone in the directory server except for
# GroupA and GroupB access to the directory “my_stuff”

  acl "path=/export/user/990628.1/docs/my_stuff/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (group = "GroupA,GroupB");
```

If someone requests the URL: `http://`*`server_name`*`/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, 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.

Restricting Access to Your Web Site

This section takes you through the process of restricting access to the files or directories 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.

You can set access control through two iPlanet Web Server mechanisms, both offer flexibility in the scope of your desired settings:

- Administration Server
- Server Manager

Note You can set access control globally for all servers through the Administration Server or for a resource within a specific server instance through the Server Manager. This section describes how to use the Server Manager to set up access control within a specific server instance. For more information regarding how to use the Administration Server to set access control globally, see "Restricting Server Access," on page 77 in Chapter 3, "Setting Administration Preferences."

There is also a section of examples you can review in the section "Access Control Examples" on page 358Access Control Examples.

To create an access control rule:

1. From the Server Manager, choose the **Preferences** tab.
2. Click the **Restrict Access** link.

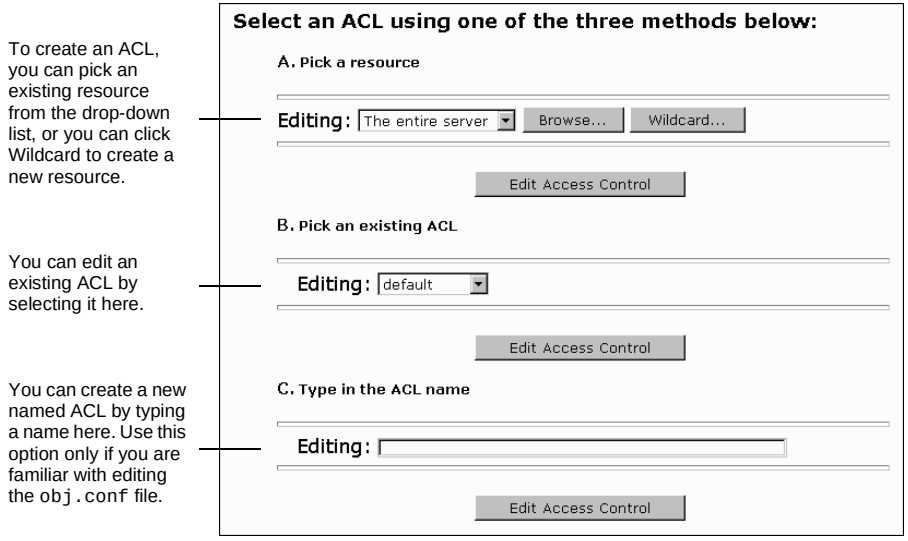
   The Access Control List Management Page appears. There are three parts to this page:

   - **Pick a resource** allows you to 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** lists all the ACLs you have enabled. Even if an ACL exists, if you have not enabled it, it will not appear in this list.

   Do not delete all the ACL rules from the ACL files. At least one ACL file is required to start the server, and the ACL file must have at least one ACL rule. If you delete all the ACL rules in the ACL files, and try to restart the server, you will see a syntax error.

   - **Type in the ACL name** allows you 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 14.2 The Restrict Access page has three sections.



3. Specify the part of the server (the resource) that you want to control in the **Pick a resource** section.

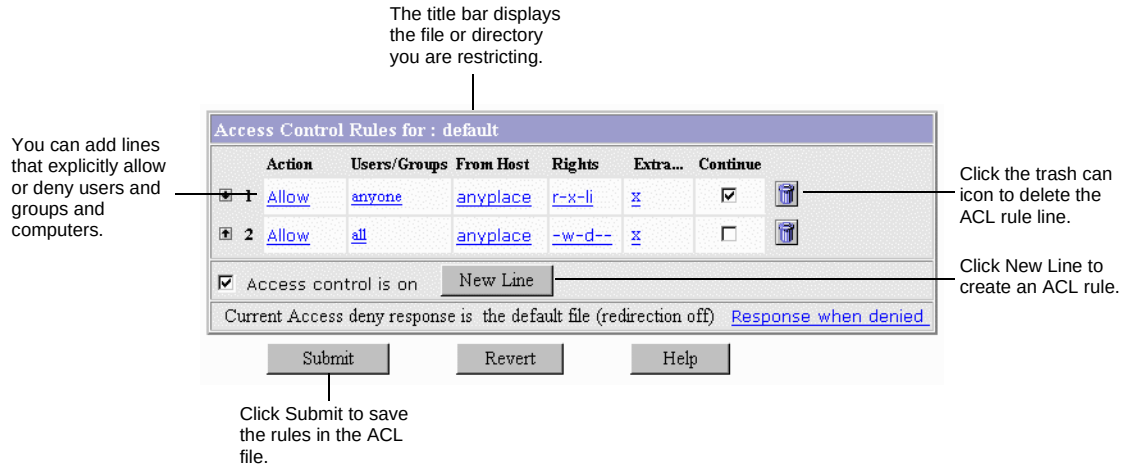
   For example, you can select Entire Server to set up access control for the entire server. The drop-down list contains an entry for each ACL resource defined in the server root.

   For some common examples of resources you might use for access control, see Table 14.1.

4. Click **Edit Access Control**.

   The page 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 ACL for iPlanet Web Administration Server, begins with two non-editable deny statements by default. The following figure briefly describes the page elements.

Figure 14.3 The ACL page contains links that, when clicked, display additional information in the bottom frame (not shown).



5. Click **New Line**.

   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.

6. Select the action you want to apply to the rule by clicking **Deny**.

   You can specify whether to deny or allow access to the users, groups, or hosts specified in the following steps in the bottom frame. Select the option you want, and then click **Update**.

7. Specify User-Group authentication by clicking "**anyone**" listed under the Users/Groups column.

   The bottom frame allows you to configure User-Group authentication. By default, there is no authentication, meaning anyone can access the server resource. Select the options you want, and then click **Update**.

8. Specify the computers you want to include in the rule by clicking **anyplace**.

   You can enter wildcard patterns of host names or IP addresses to allow or deny in the bottom frame. Select the options you want, and then click **Update**.

9. Specify the access rights you want to include in the rule by clicking **all**. Select the access rights in the bottom frame, and then click **Update**.

10. Specify the programs you want to restrict. Programs are the forms in the Server Manager for the server you selected. For example, you can restrict access to all forms for configuring the administration server by checking the "All Programs" radio button. If you want to restrict access to one or two sets of forms, choose the categories in the drop-down list. If you want to restrict access to one form in a category, type the name of the form in the "Program Items" field. For example, to restrict access to the access control form, type `distacl` in the Program Items field. For more information, see "Access to Programs" on page 354.

    Click **Update** to add the programs options to the rules for the line you're editing.

11. 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.

12. Select **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.

13. Repeat steps 5 through 11 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.

14. Click **Submit** 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 page. 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.

Table 14.1 LDAP Attributes

Resource wildcard	What it means
default	A named ACL created during installation that restricts write access so only users in the LDAP directory can publish documents (for example, by using the web publisher).
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/netscape/server4/docs/cgi-bin/*`	Controls access to all files and directories in the `cgi-bin` directory. You must specify an absolute path. On NT, the path must include the drive letter.
`uri="/sales"`	Controls access to the `sales` directory in the document root. To specify URIs, create a named ACL.

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

Setting Access Control Actions

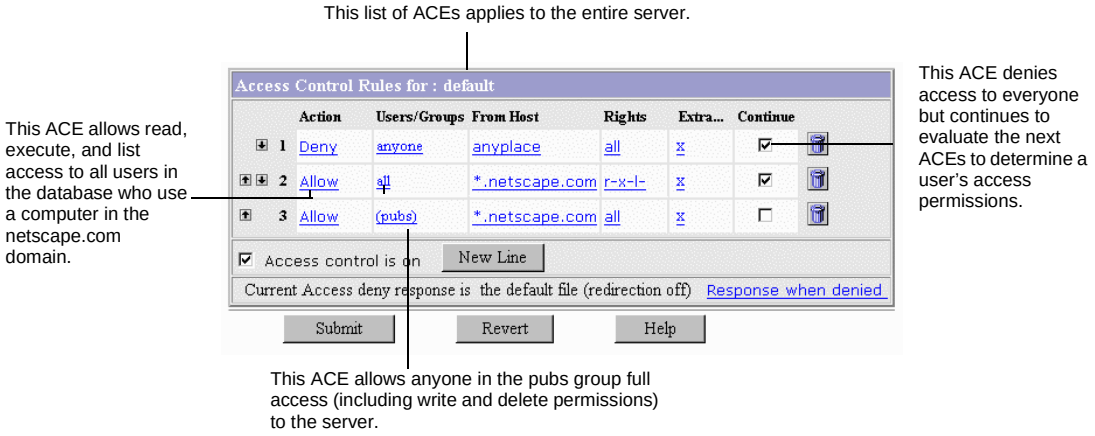
You can specify the action the server takes when a request matches the access control rule.

- **Allow** means the users or systems can access the requested resource.
- **Deny** means the users or systems cannot access the resource.

The server goes through the list of ACEs to determine the access permissions. 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 14.4 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 14.4You can combine Deny and Allow statements in an ACL.



Specifying Users and Groups

You can restrict access to the Administration Server or 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.

iPlanet Web Server uses a list of users, who might be sorted into groups, to determine access rights for the user requesting a resource. You must define an administrators group (the group you set up for distributed administration) for access control in the Administration Server. The list of users (and the groups they are included in) are stored in an LDAP server, such as Netscape Directory Server. You should make sure the database contains users and groups (including the administrators group) 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 additional options appear in the bottom frame. The following list describes the options in the bottom frame.

- **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. For the Administration Server, this means that anyone in the administrators group that you specified with distributed administration can access the pages.
- **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**." For the Administration Server, the users you specify must also be in the "administrators" group you specified for distributed administration.
- **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. For the Administration Server, the users in the groups you specify must also be in the "administrators" group you specified for distributed administration.
  - **User** matches the individual users you specify.
- **Prompt for authentication** lets you specify message text that appears in the authentication dialog box. 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, 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 an LDAP directory. 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 choose the database you want to use in the ACL by selecting it in the drop-down list. If you use the access control API to use a custom database (for example, to use an Oracle or Informix database), you can type the name of the database in the "Other" field in the User/Group window.

Specifying Host Names and IP Addresses

You can restrict access to the Administration Server or 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 `*.iplanet.com`. You can set different hostnames and IP addresses that the superuser must use when accessing the Administration Server.

To specify users from hostnames or IP addresses, follow the directions for restricting access in "Restricting Access to Your Web Site" on page 344. When you click the From Host field (the link called **anyplace**), additional options appear 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 not `198.*.251.30`.

For hostnames, the `*` must also replace an entire component of the name. That is, `*.iplanet.com` is acceptable, but `*sers.iplanet.com` is not. When the `*` appears in a hostname, it must be the left-most character. For example, `*.iplanet.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 choose 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. To delete a file, a user must have both write and delete privileges.
- **Execute** access applies to server-side applications, such as CGI programs, Java applets, and agents.

- **Delete** access means a user who also has write privileges 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.

Access to Programs

You can select areas of the administration server that administrators can access. You can choose groups of tabs that appear in the Server Manager (such as Cluster Management), or you can choose specific pages that appear as links in the left frame of the Server Manager (such as "New User" in the User & Groups tab).

To control access to a program in a server, perform the following steps:

1. From the Administration Server, choose the **Global Settings** tab
2. Choose **Restrict Access**.
3. From the drop-down list, choose the server whose administration access you want to restrict. The administration server is labeled "`https-admserv`." Other servers are labeled with their type and their server id (for example, `https-mozilla`).

   When you select a server to restrict, you are restricting who can view the Server Manager pages and which pages they can use to configure that server. For example, you might allow some administrators to configure the Users & Groups section of the administration server and not allow them access to the Global Settings.
4. Click **Edit ACL**. The web server displays the two-frame access control pages.
5. Each ACL begins with two deny lines (the default setting), one that restricts access to only those users in the "administrators" group set for distributed administration, and another that restricts access to all users. If you want to

change either of these lines, you need to manually edit the ACL file. Click **New Line** to add a rule to the ACL. Each rule you create allows access to the server. By specifically allowing access for users, you reduce the risk that you'll allow access to users you don't want.

6. Choose the users, groups, hosts, and IP addresses you want to apply to this access control rule.

7. By default, administrators have access to all programs for a server. Click the **All** link under **Programs** in the top frame. The bottom frame displays a page that lists the programs for the server type you selected.

8. Select **Only the following**, and then select the Program Groups you want to apply to the rule. You can choose multiple groups by pressing the Control key and then clicking the groups you want.

   The Program Groups listed use the same name as the buttons in the top frame of the Server Manager for the server type you selected. For example, in the administration server, there are tabs labeled Preferences, Global Settings, and so on. When an administrator accesses the administration server, the server uses their username, host, and IP to determine what pages they'll see. If they have access to only one or two pages, they will only see those pages.

9. You can control access to a specific page within a tab. Type the name of the page in the **Program Items** field. To determine the name of a page, place your pointer over the link in the left frame of the Administration Server and then view the text in the status bar on the bottom of your browser. The last word after the last %2b is the name for that page.

   For example, suppose you have one person who administers a Netscape Directory Server and you want that person to have access only to the "Configure Directory Service" page. In this case, you would set up a rule that applies to them (host, IP, and so on), and then you would enter `dsconfig` in the Program Items field.

10. Click Update and then Submit to save the access control rule.

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 "ACL File Syntax" on page 466 and "Referencing ACL Files in obj.conf" on page 472.

Selecting "Access control on"

When you uncheck the option labeled "Access control on," 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.

If you want to deactivate an ACL, you can comment out the ACL lines in the file `generated-https-server-id.acl` by putting # signs at the beginning of each line.

From the Administration Server, you could create and turn on access control for a specific server instance and leave it off (which is the default) for other servers. For example, you could deny all access to the Server Manager pages from the Administration Server. With distributed administration on and access control off by default for any other servers, administrators could still access and configure the other servers, but they cannot configure the Administration Server.

Note This access control is in addition to the user being in the administrators group set for distributed administration. The the Administration Server first checks that a user (other than superuser) is in the administrators group, and then it evaluates the access control rules.

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 was not found (the HTTP error code 404 Not Found is also sent).

To change what message is sent for a particular ACL, perform the following steps:

1. In the ACL page, click the **Response when denied** link.
2. In the lower frame, check the **Respond with the following file** radio button.
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. The server must have read access to this file, so you should consider putting the file in the document root.

   Make sure the file does not contain references to other files or images because they will not be sent.
4. Click **Update**.

   Make sure any users who get the response file have access to that file. 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.
5. Make sure you submit the access control rule by clicking **Submit** in the top frame.

Access Control Examples

This section describes some common examples for restricting access to a web server and its contents. Some of these examples assume you set up the "default" ACL to deny anyone access to the server. You can also add a "deny all" line as the first rule to each of these examples, as done in the example for the entire server (see "Restricting Access to the Entire Server" on page 358).

This section includes the following topics:

- Restricting Access to the Entire Server
- Restricting Access to a Directory (Path)
- Restricting Access to a URI (Path)
- Restricting Access to a File Type
- Restricting Access Based on Time of Day

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, perform the following steps:

1. In iPlanet Web Server, choose **Server Preferences**.
2. Click the **Restrict Access** link.

   The web server displays the Access Control List Management page.
3. In the section called **Pick a Resource**, select "The entire server" from the Editing drop-down list and then click **Edit Access Control**.

   The two-frame page appears.
4. 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.

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

6. Click the "**anyone**" link in the second rule. In the bottom frame, 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**.

7. Click the "**anyplace**" link in the second rule. In the bottom frame, type a wildcard pattern for the host names of the computers you want to allow.

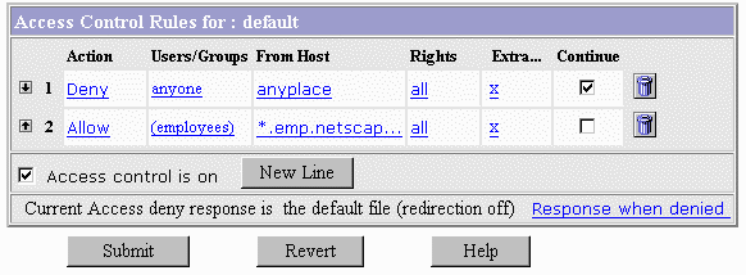
   For example, type `*.emp.mozilla.com` in the Host Names field. Click **Update**.

8. Unselect **Continue** in the top frame, and then click **Submit**.

   The frame should look like the one in Figure 14.5.

9. Submit your changes.

Figure 14.5 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";
authenticate (user,group) {
  prompt = “Web Server”
}
deny (all)
  user = "anyone";
allow absolute (all)
  (group = "employees") and
  (dns = "*.emp.netscape.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, perform the following steps:

1. In the Server Manager, choose **Server Preferences**.
2. Click the **Restrict Access** link.

   The web server displays the Access Control List Management page.
3. In the section called **Pick a Resource**, click the **Browse** button.

   In the page that appears, click the link for the directory you want to restrict. The directories listed in this page 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 **Options** and check the box labeled **List files as well as directories** and then click OK.

4. Click **Edit Access Control**.

The two-frame pages appear.

5. 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.

6. Click **Deny** in the second rule. In the bottom frame that appears, check Allow, and then click **Update**.

7. Click **anyone** in the second rule. In the bottom frame, type the group you want to have access to the server. For this example, type `executives` in the Group field. Click **Update**.

8. Click **all** 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**.

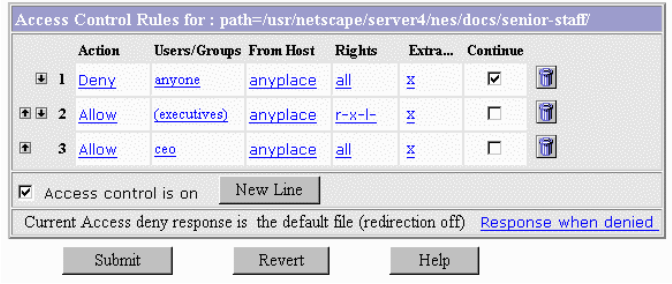
9. Click **New Line** to create a rule for the "ceo" user. Check **Allow** for the third rule.

10. Click **anyone**. In the bottom frame, type `ceo` in the User field. Click **Update**.

11. 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 3. The frame should look like the one in Figure 14.6.

Figure 14.6 Restricting access to a path in the server



12. Click **Submit** and save and apply your changes.

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

```
acl "path=/usr/netscape/server4/nes/docs/senior-staff/";
deny (all)
    user = "anyone";
allow absolute (read,execute,list)
    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, perform the following steps:

1. In the Server Manager, choose **Server Preferences**.
2. Click the **Restrict Access** link.

   The web server displays the Access Control List Management page.
3. In the **Type in the ACL name** section, type the URI you want to control. For example, type `uri=/my_directory`.
4. Click **Edit Access Control**.

   The two-frame pages appear.
5. Click **New Line** to create the first rule that allows all users read access.

6. Click the **Deny** link. In the bottom frame that appears, check **Allow**, and then click **Update**.

7. Click the **all** link in the top frame. Uncheck the **Write and Delete** access rights.

   This means users cannot add or remove files, but they can view them and run any applications in the directories. Click **Update**.

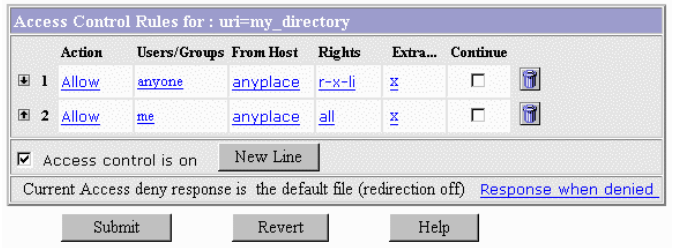
8. Click **New Line** to create a rule for the owner of the directory. Check **Allow** for the second rule.

9. Click **anyone**. In the bottom frame, type `me` in the User field. Click **Update**.

10. 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 3. The frame should look like the one in Figure 14.7.

Figure 14.7 Restricting access to a URI (path) in the document root



11. Click **Submit** and save and apply your changes.

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, perform the following steps:

1. In the Server Manager, choose **Server Preferences**.
2. Click **Restrict Access**.

   The web server displays the Access Control List Management page.
3. In the **Pick a resource** section, click **Wildcard**. In the prompt that appears, type `*.cgi` and click OK.

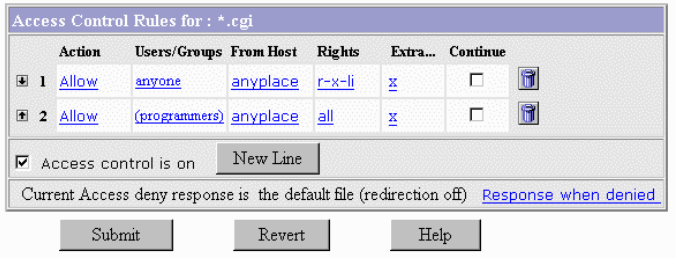
   This wildcard pattern matches any request that contains a file or directory with the `.cgi` extension.
4. Click **Edit Access Control**.

   The two-frame pages appear.
5. Click **New Line** to create the first rule that will allow all users read access.
6. Click **Deny**. In the bottom frame that appears, check **Allow**, and then click **Update**.
7. Click **all** 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**.
8. Click **New Line** to create a rule that allows write and delete access to the "programmers" group. Check **Allow** for the second rule.
9. Click **anyone**. In the bottom frame, type `programmers` in the **Group** field. Click **Update**.

   The frame should look like the one in Figure 14.8.

Figure 14.8Restricting access to a file type—in this case, to files with the .cgi extension



10. Click **Submit** and save and apply your changes.

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 web server checks ACLs in the following 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, perform the following steps:

1. In the Server Manager, choose **Server Preferences**.
2. Click **Restrict Access**.
3. In the **Pick a Resource** section, select "The entire server" from the Editing drop-down list. (You can select any resource.) Click **Edit Access Control**.

   The server displays the two-frame pages.
4. Click **New Line**.
5. Click the **Deny** link. In the bottom frame 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 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**.
7. Click **New Line** to create a rule that restricts the write and delete methods. Check **Allow** for the second rule.
8. Click the **X** link to create a customized expression. In the bottom frame, 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 frame will display "Unrecognized expressions" in the Users/Groups and From Host fields because you created a custom expression.

9. 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 effect.

Access Control For Web Publishing

Web Publisher users can control who accesses their Web Publisher documents and directories and what operations different users or different groups of users can perform upon the files. They can also completely prohibit access to a file or folder or you can restrict access to certain authenticated users.

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.

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 Restrict Access function (choose Server Preferences and click the Restrict Access link) 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.

Note For Unix/Linux, if you expect web publishing users to publish documents to a directory, you need to set the Unix/Linux file permissions to give them write access to that directory. You should also disable write permissions for directories you do not want them to publish to.

Web Publisher has many operations that are restricted to the broad category of valid server user. Many ACL rules, such as that for agents services, simply require a user to be valid for the server. That is, users who are defined in the server's users database.

When you start Web Publisher, you are immediately prompted with the user name authorization dialog box. You can leave this blank and operate as an anonymous user, but as soon as you attempt to perform an operation restricted to a valid user, you are again prompted for your user name. At this point, you are also asked to enter your password, and only authenticated users can continue with the operation.

Ownership of Files and Folders

Web Publisher files and folders can be owned by individual users. Only the owner of a file or folder can define its access control definitions or reassign its ownership. If a file or folder has no owner, no one can modify its access control. You can define an access control definition that restricts certain operations to the user who is the current owner of a file or folder, even when ownership is reassigned.

Your can do a bulk ownership assignment for your Web Publisher users, and the users can assign ownership for an individual file or folder through the Web Publisher properties page, or they can become the owner of a file or folder as a result of an automatic assignment by Web Publisher when you perform certain actions.

Chapter

15

Configuring Web Publishing

iPlanet Web Server clients can use 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, and access control.

Note The Web Publisher function is not available on Linux platforms.

This chapter contains the following sections:

- Using Netshare
- Setting Access Control For Web Publisher Owners
- Indexing and Updating Properties
- Changing the Web Publishing State
- Maintaining Web Publishing Data
- Unlocking Files
- Adding Custom Properties
- Managing Properties
- Customizing Your Netshare Home Page
- Customizing the Web Publisher User Interface

Using Netshare

Netshare provides an iPlanet Web Server user with a personal home page from which they can store, share, and manage their server documents. Netshare is a convenient starting point for using the iPlanet Web Server user services: Web Publisher and search. From their home page, users can also obtain information about how they are defined in the server's user directory, such as their name, password, and telephone extension.

When you create a Netshare home directory for a user, the user is assigned as the owner of the directory and all its files. By default, only the owner can write to the directory although other users can read the files. Others cannot make any changes to the files unless the owner explicitly provides such access permissions.

Once you've created a Netshare home directory for a user, they can access the default Netshare home page. To access a user's default Netshare home page, the user types in the following URL:

`http://`*`ServerID`*`/netshare/`*`UserID`*

After authentication, the user's default Netshare home page is displayed with a set of links to many server functions:

Table 15.1 Netshare Home Page Links

Link Name	Description
Web Publisher	Users have direct access to the files and folders in their home directory.
Access control	Users can restrict access to their home directory.
Search	Users can search on any collection set up for their server.
User info	Users can review and modify their user information.
User's Guide	Users can look at Netshare online help.

This section includes the following topics:

- Setting Up the Server and Creating Netshare Home Directories
- Before You Start

- Using the Server Manager
- Accessing the Web Publisher Home Page

Setting Up the Server and Creating Netshare Home Directories

As the server administrator, you need to configure Netshare for your server and for your server's users before they can use Netshare. Once you have set up Netshare, you need to create a Netshare home directory for any user or group who wants to use Netshare. Netshare provides an interface for server administrators: the iPlanet Web Server, Server Manager user interface (Set Up Netshare and Create Netshare).

Before You Start

Before you set up Netshare for your users, you need to be sure that the Web Publishing functions are turned on for your server, that you understand how Netshare's default naming conventions operate, what the configuration file contains, and what it means to mark a user as licensed.

Server Features That Must Be Enabled

In order to use the functions of Netshare fully, including Web Publisher and search, each of these functions must be turned on for your server. By default, they are all not enabled, but you may wish to verify their state.

To turn on the Web Publisher, use The Web Publishing State Page in the Server Manager.

To check the state of Web Publisher, use The Search State Page in the Server Manager.

Note If Search is turned on before Web Publishing then the default collection is not created until after a force index is performed. This happens only if Web Publishing is enabled after Search. The reason that the Web Publishing collection does not show up in search is that at the time the search init is run, Web Publishing has not been created. If you restart the server, then it will show up correctly.

Netshare Directory Naming Conventions

To facilitate handling large quantities of individual home directories for every Netshare user, a naming convention has been defined. Its page is at *doc_root/netshare_directory/home_directory*. The default is to use the primary document directory for your server (*server*/docs), to use /netshare as the Netshare directory, and to use the user's User ID as the home directory's name. Thus, on a default Windows NT installation for the user jdoe, the Netshare user's home directory would created be at C:\Netscape\server4\docs\netshare\jdoe.

As server administrator, you can select one of the additional document directories defined for your server as the document root, and you can define different directories for the Netshare and home directories. If you change these values on the iPlanet Web Server Set Up Netshare page, the configuration file is changed to reflect your changes, and all home directories added subsequently use the new directory values. If, however, you use the Netshare utility to indicate different directories, the configuration file is not changed, so only the user directories currently being added are affected.

One situation in which you might want to change the default directory path is when you want to create a home directory for a user that does not map to their user ID. For example, the user jdoe wants an additional Netshare directory called Project1. In this case, the user ID would be jdoe and jdoe would be assigned as owner of the Project1 directory.

The Netshare Configuration File

Netshare uses a configuration file, netshare.conf, that contains the following data:

- Document root
- Netshare parent directory
- Template filename

You can only modify this file from the Set Up Netshare page or manually through a text editor. When you use the Netshare utility, the configuration file is not affected.

When you use the Set Up Netshare page to change a default directory or template file, you are updating the values in the configuration file. From then on, any home directory that you create uses the new values.

Marking Users As Licensed

In order to create a Netshare home directory for a user, the user must be marked as having been granted a Client Access License for an iPlanet Web Server.

Note After your server is installed and before being able to mark a user as licensed, you need to go to the Set Up Netshare page (choose **Web Publishing** and clicking the **Set Up Netshare** link). Displaying this page causes an essential modification to an internal configuration file. You need only do this one time after the server is installed.

For an existing user, mark the licensing in one of these ways:

- as part of creating an individual Netshare home directory through the Server Manager
- as part of creating an individual Netshare home directory through the Netshare utility
- by performing the following steps:

1. From the Administration Server, choose the **Users & Groups**.
2. Click **Manage Users**. The web server displays the Manage Users page.
3. Click **Find**.
4. On the page listing all matched user IDs, click the desired user ID.
5. Click **Licenses**.
6. Select **iPlanet Web Server** and click **Save Changes**.

Access Control For Netshare

When you create a Netshare home directory for a user ID, the server assigns the user as its owner and the user is the only one who can write to the directory. Other users can read the user's files, but cannot make any changes to them unless the user explicitly provides such access permissions.

The default access permission is to allow anyone defined as a valid server user to read any Netshare directory, but only the designated owner of the Netshare home directory can modify the files.

This is the default ACL that is applied to the Netshare parent directory:

```
allow (all) user = 'owner';
```

When you create a Netshare home directory for a group, the server assigns ownership of the files and folders in that directory to the owner's user ID. This also gives all users in the group read-write access permission for all files and folders in the home directory. Because this requires creating a new ACL rule for this particular group, this forces the server to restart to pick up the new ACL information.

Using the Server Manager

The iPlanet Web Server provides pages that allow you to modify Netshare configuration settings (Set Up Netshare) and create Netshare home directories (Create Netshare).

The Set Up Netshare Page

The Set Up Netshare page allows you to modify the Netshare configuration settings. You can change some of the Netshare configuration information for your server and the configuration file is updated with your changes. When you have set up Netshare for your server, you can create Netshare home directories for your users.

To change the values in the configuration file, use The Set Up Netshare Page in the Server Manager.

The Create Netshare Page

You can use the Create Netshare page to create Netshare home directories for an individual user, for a specified group, or for all users who have been marked as licensed. The last choice is particularly useful for server administrators who wish to add Netshare home directories for all existing users.

Note You must have already set up Netshare for your server before you can create Netshare home directories and Web Publishing must be enabled before your users can use Netshare.

To Create a Netshare Home Directory for a Single User

To create a Netshare home directory for a single user, perform the following steps:

1. From the Server Manager, choose **Web Publishing**.
2. Click the **Create Netshare** link.

   The web server displays the Create Netshare page.
3. Click **A single user**.
4. Type the user's user ID.
5. Ignore the Owner field.
6. (Optional) Enter another Netshare home directory if you do not want to accept the default of using the user ID.
7. Click the **Create** button.

   This marks the user as licensed if not yet marked as such, creates the user's Netshare home directory, and assigns ownership of the files and folders in that directory to the user ID that was specified. If you attempt to create a home directory that already exists, you receive an error message.

To Create a Netshare Home Directory for a Group

To create a Netshare home directory for a group, perform the following steps:

1. From the Server Manager, choose **Web Publishing**.
2. Click the **Create Netshare** link.

   The web server displays the Create Netshare page.
3. Click **A group**.
4. Enter the name of the group.
5. Enter the name of the group's owner.

   This name must be a valid member of the group.

6. (Optional) Enter another Netshare home directory if you do not want to accept the default of using the group name.

7. Click the **Create** button.

   This restarts the server because you have added a new ACL, granting special access to the members of the group. When you add a single user's Netshare, the default ACL is sufficient.

   This marks the group's owner as a licensed user if not yet marked as such, creates the group's Netshare home directory, and assigns ownership of the files and folders in that directory to the owner's user ID. This also gives all users in the group read-write access permission for all files and folders in the home directory. If you attempt to create a home directory that already exists, you receive an error message.

To Create a Netshare Home Directory for All Users

To create Netshare home directories for all users at once, follow these steps:

1. From the Server Manager, choose **Web Publishing**.

2. Click the **Create Netshare** link.

   The web server displays the Create Netshare page.

3. Click "All users marked as licensed."

4. Click Create.

This searches through the LDAP user directory for your server and creates a Netshare home directory for each user who has been marked as licensed but who does not yet have a Netshare directory. If you attempt to create a home directory that already exists, you receive an error message.

Note You can use the Netshare utility (the `-l` option) to perform a batch update, marking all users as licensed, before using this page.

Accessing the Web Publisher Home Page

When you have configured web publishing for your server, you and your users can access the Web Publisher home page at the following URL:

```
http://ServerID/publisher
```

This displays the home page, which provides a link for downloading the web publishing plug-in appropriate for your browser and the Start Web Publisher button, which launches the Web Publisher applet. The page also includes a set of links to a QuickStart tutorial and to the entire Web Publisher online help system, *Netshare & Web Publisher User's Guide*.

The online help system is also available through user components such as search and Web Publisher. To access the help system, use the Help menu command in Web Publisher, or click Help on the search interface pages or on the Web Publisher Services page.

Setting Access Control For Web Publisher Owners

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 Index and Update Properties page (choose Web Publishing and click the Index and Update Properties link) or through Web Publisher. The Index and Update Properties page 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 list (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.

Note If you change the owner of the Netshare directory and all subsequent subdirectories, then only owner can write to these directories. If you change the Netshare root owner, but not the owner of the subdirectories, then the owners of the subdirectories can still write within the directories. It is important to note that if you change the name of a file (with this specific ACL settings), then the user needs to be the owner of both the enclosing directory, and the file itself.

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 Restrict Access function (choose Server Preferences and click the Restrict Access link) to create an ACL that is more appropriate for web publishing.

For example, you could create an ACL as shown in the following example:

```
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.

For more information about setting access control, see “Restricting Access to Your Web Site,” on page 344 in Chapter 14, “Controlling Access to Your Server.”

Note For Unix/Linux, if you expect web publishing users to publish documents to a directory, you need to set the Unix/Linux file permissions to give them write access to that directory. You should also disable write permissions for directories you do not want them to publish to.

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 page 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 page (choose Web Publishing and click the Index and Update Properties link) 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.

Note Using this function clears the link status database of all current link checking information. You must recheck your links after indexing files.

To index and update properties, perform the following steps:

1. From the Server Manager, choose **Web Publishing**.
2. Click **Index** and **Update Properties**.

   The web server displays the Index and Update Properties page.
3. Choose the path of the directory to be indexed.

   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 page and the directory name appears in the Document Directory field.

Note You cannot use this function to index files that are larger than 3MB in size. You can, however, do an automatic indexing of such large files through the Property Sheet in Web Publisher (through the Web Publisher View Properties menu command) by checking "Make contents searchable".

Unix/Linux. You can index the contents of your users' files and folders that are in their default user home directories as defined by the Content Management | User Document Directories function. For example, if user document directories are active on your server and the default ~USERNAME/`public_html` has been defined for your server, that entry is displayed as one of the permitted document directories you can index. This indexes all user document directories that exist currently on your server according to the criteria you select in the Index and Update Properties page.

4. If you also want to index the subdirectories within the specified directory, click **Include Subdirectories**.

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 the following pattern (complete with parentheses) to index all HTML documents:

   ```
   (*.htm|*.html)
   ```

   You can define multiple wildcards in an expression. See Chapter 2, "Administering iPlanet Web Servers," for details of the syntax for wildcard patterns.

6. If this is the first time you index web publishing documents, check **Index unindexed documents**.

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 select **Set document owner to** 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 **Index document contents**.

   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 page. The information is also logged to the `yourServer/plugins/content_mgr/logs/wpsimport.log` log file. New data is appended to the log, so you may want to monitor its size as you proceed through many indexing operations.

You can enable logging in the indexing engine in two ways:

- To get information on all the metadata being indexed, in the `[NS-loader]` section of `webpub.conf` set `NS-debug-bulk=Y`. Then a file named `Bulk.Save` is created under the *`server_root/`*`https-`*`server-id`*`/search/collections/web_htm/idx/` directory. This file logs all the data that have been indexed.

- For Verity, you can append "`;127`" to include the `NS-initparms` in the `[NS-search]` section of `webpub.conf` and set `NS-debug-log=Y` in the `[NS-loader]` section of `webpub.conf`. This ensures that maximum logging is generated from Verity, and the log messages go into *`server_root`*`/https-`*`server-id`*`/logs/nsloader.log`.

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 activate or deactivate web publishing using The Web Publishing State Page in the Server Manager.

Warning If your server is outside the firewall, you should turn off directory indexing (from the Server Manager, choose Content Management, click the Document Preferences link, and click None in the Directory Indexing section) as well as web publishing. This ensures that your directory structure, file names, and web publishing features are not accessible.

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 sync. 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.

   The web server displays the Maintain Web Publishing Data page.
3. Choose the directory that contains the web publishing data to be maintained.

   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 page and the directory name appears in the Document Directory field.
4. To also report on or repair the subdirectories within the specified directory, click **Include Subdirectories**.
5. To report on the collection, click **Report**. This reports on the selected document directory.
6. To repair the collection, click **Repair**.

   This repairs inconsistencies in the selected document directory.
7. To optimize the collection, click **Optimize**.

   This optimizes the entire web publishing collection.

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 Product Name goes here as part of an edit or download operation.

For further information about locking and unlocking files in Web Publisher, access the online *Netshare & Publisher User's Guide* through the Help menu command in Web Publisher, or the Help button on the search interface, or the Web Publisher Services page.

Be cautious in using this function because by unlocking a file that was locked, you are making the file 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, perform the following steps:

1. From the Server Manager, choose **Web Publishing**.
2. Click the **Unlock File** link.

   The web server displays the Unlock File page.
3. Select the file or directory you want to unlock.

   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 page 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 You cannot use this page to unlock a file that begins with a period (as in `.cshrc`), a plus (+), an equals sign (=), an ampersand (&), or any hexadecimal character. You 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 *Netshare & Web Publisher User's Guide* through the Help menu command in Web Publisher, or the Help button on the search interface, 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 properties 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.

This means that if you extend the maximum settings to add additional attributes, you cannot automatically use the new attributes in the existing web publishing collection. To allow this, 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 to automatically create a new web publishing collection for you. (The `link_mgr` collection is an internal file that's part of web publishing.)

To add a custom file property, perform the following steps:

1. From the Server Manager, choose **Web Publishing**.
2. Click the **Add Custom Properties** link.

   The web server displays the Add Custom Properties page.
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 contain an underscore.

4. Select the property's type from the **Property Type** 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. See "Configuring Files Manually" on page 408 in Chapter 16, "Using Search," for details on how to change the `webpub.conf` file.

5. Select one of the **Permissions** buttons.

6. Click either **Read only** or **Modifiable**. By default, this selection 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.

7. 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.

8. If the property you are adding is actually an HTML file attribute that has been tagged with the HTML META tag, click HTML META.

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.

9. 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.

This section includes the following topics:

- To Manage File Properties
- To Remove a Custom Property
- To Edit a Custom Property

To Manage File Properties

To manage file properties, perform the following steps:

1. From the Server Manager, choose **Web Publishing**.
2. Click the **Manage Properties** link.
3. The web server displays the Manage Properties page, which subsequently displays all available properties.

To Remove a Custom Property

To remove a custom property, perform the following steps:

1. Click the **Remove** link for the property.

   The web server displays the Remove Custom Property page.

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

To edit a custom property, perform the following steps:

1. Click the **Edit** link for the property.

   The web server displays the Edit Custom Property page.

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.

Customizing Your Netshare Home Page

By default, a user's Netshare home page displays the `netshare.html` file in the right frame. Initially this HTML file contains mostly text and a few sample links, but you or your users can revise this file to contain other text, graphics, links, and other HTML elements.

This file is the starting point for a user's workspace on the remote server and is what other users first see when they access the user's home page. You or the owner of the home page may want to provide some explanation of what other files, folders, and services are available through the home page and display some navigational links to route other users through the site.

These are the default files that are installed in a Netshare home directory:

- `netshare.html` - The default text that appears in the right frame.
- `banner.html` - The banner across the top.
- `home.html` - The frameset itself.
- `menu.html` - The set of links in the left frame.
- `test1.html` - A sample file.
- `test2.html` - A sample file.

Customizing the Web Publisher User Interface

Web Publisher uses a standard set of default properties to describe its files and folders. These properties are listed in the Manage Properties page (choose Web Publishing and click the Manager Properties link) and are used in the HTML pages that the Web Publisher user sees.

As server administrator, you can customize these pages to meet specific user requirements. These pages are defined as a set of modifiable pattern files that contain pattern variables for the Web Publisher properties. These variables are named by taking the attribute name defined in the `dblist.ini` file (located in the `yourServer/plugins/search.admin directory`) and adding the

prefix $$. For example, you can $$ to the variable `CM_LOCK_OWNER` to create the `$$CM_LOCK_OWNER` variable for displaying the lock owner in an HTML pattern file.

The Web Publisher Attributes

To understand how these work, look at the `dblist.ini` file that came as part of the default installation of your server. You can see there a series of attributes called `NS-idxattr1` through `NS-idxattr27`. These are the default Web Publisher attributes and they follow this standard syntax:

```
NS-idxattrn=CM_attributeName;displayName;TYPE;size
```

where

- *n* = attribute number
- *attributeName* = internal attribute name
- *displayName* = name displayed in Manage Properties page
- *TYPE* = TXT (text), NUM (numeric), or DAT (date)
- *size* = size of field Web Publishing attributes, listed in order of `NS-idxattr` number

Table 15.2 Web Publishing attributes. listed in order of NS-idxattr number

No.	Attribute Name	Display Name	Type	Size	Description
1	CM_PPATH	File-Name	TXT	25	The physical path name of the file or folder that's being operated upon.
2	CM_MDATE	Modified-Date	DAT	9	The date of the last modification to the file or folder.
3	CM_CDATE	Creation-Date	DAT	9	The date when the file or folder was created
4	CM_SIZE	Size	NUM	9	The size in bytes.
5	CM_ID	ID	NUM	11	An internal ID.
6	CM_RES-STAT	Resource-Stat	TXT	11	An internal status.

Table 15.2 Web Publishing attributes. listed in order of NS-idxattr number

No.	Attribute Name	Display Name	Type	Size	Description
7	CM_URI	URI	TXT	9	The URI of a file or folder.
8	CM_OWNER	Owner	TXT	9	The owner of a file or folder.
9	CM_COUNTER	Counter.	TXT	9	An internal counter.
10	CM_VERSION	Version	TXT	9	The number for a particular version of a file.
11	CM_LINK_STAT	Link-Status	TXT	9	A flag indicating the state of the links in a file or folder. A link can be working, broken, or external (file) or unchecked (folder).
12	CM_LOCK_STAT	Lock-Status	TXT	9	A flag indicating whether the file of folder is locked.
13	CM_LOCK_OWNER	Lock-Owner	TXT	9	The user who locked the file manually or who is editing the file.
14	CM_RECENT_AUTHOR	Modified-by	TXT	9	The user who made the most recent modification to the file or folder.
15	CM_RECENT_COMMENT	Most-Recent-Comment	TXT	9	The most recent comment added for a file as part of an upload or publish operation.
16	CM_VERSIONED	Versioned	TXT	9	A flag indicating whether the file is under version control.
17	CM_AUTHOR	Author	TXT	13	The author defined for the file with the HTML META tag of Author.
18	CM_DESCRIPTION	Description	TXT	13	The text defined for the file with the HTML META tag of Description.
19	CM_TITLE	Assigned-Title	TXT	8	The title defined for the file with the HTML META tag of Title. (not used)
20	CM_LOCALE	Locale	TXT	9	The language defined for Web Publisher.

Table 15.2 Web Publishing attributes. listed in order of NS-idxattr number

No.	Attribute Name	Display Name	Type	Size	Description
21	CM_IS_INDEXED	Is-indexed	TXT	9	A flag indicating that the content and metadata of a file has been indexed.
22	CM_IS_PERSISTENT	Is-persistent	TXT	9	A flag indicating that the metadata only of a file has been indexed. The file's content has not been indexed.
23	CM_RES_TYPE	Resource-type	TXT	9	The default file extension defined for a particular file type, such as .doc for Word files and .pdf for Acrobat files.
24	CM_INDEX	Index	TXT	9	A flag indicating that the content of a file should be indexed when the metadata is indexed next.
25	Title	Title	TXT	9	The title defined for the file with the HTML META tag of Title.
26	CM_SourceType	SourceType	TXT	9	An internal indicator.
27	CM_DOC_FN	DocFileName	TXT	11	An internal filename.

The Web Publisher Pattern Files

The Web Publisher pattern files use the Web Publisher pattern variables to display values, and to pass values between the user's system and the remote server. The pattern files use a combination of HTML syntax and JavaScript to define additional variables and to display information to the user.

The following default Web Publisher pattern files are in the *yourServer*/`plugins/content_mgr/ui/text/en` directory on an English language server:

- `main.pat` - displays the 3-paned Web Publisher Services interface
- `toc.pat` - displays the left-hand set of links and buttons displayed for a file

- `dirtoc.pat` - displays the left-hand set of links and buttons displayed for a folder (version history is omitted)
- `sys-prop.pat` - displays the properties page for a file
- `dirps.pat` - displays the properties page for a folder
- `version.pat` - displays the version history page for a file
- `links.pat` - displays the check links page for a file
- `dirlink.pat` - displays the check links page for a folder
- `usrps.pat` - displays the Custom Properties page

The remaining pattern files display a short message, typically as a result of not being able to satisfy the request.

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.

There are three kinds of Web Publisher pattern variables:

- variables defined in the configuration file, `dblist.ini`, as an index attribute (`NS-idxattr`)
- variables defined as pointers to other pages, with "-NS" suffix
- variables defined internally by Web Publisher

Most modifications to the Web Publisher pattern files involve simply changing which attributes you want displayed in the properties pages, adding or removing variables from the pattern file. That is, adjusting the use of the variables defined in the `dblist.ini` file. Other modifications are likely to involve the pointer variables, identified by their "-NS" suffix. The Web Publisher-defined variables are not intended for general use, and thus are not described at length here.

Pointing Pattern Variables

There are some pattern variables that point at specific files and displays them in one of the frames in the browser.

The pointer variables that you can use in your pattern file are listed in Table 15.3.

Table 15.3 Pointing pattern variables

Variable name	Result
$$CM_CUSTOM_FEILD_NS	Custom properties page (gets data for it)
$$CM_HTML_REND_NS	The right frame displays the HTML version of the file
$$CM_LINK_INFO_NS	Link status page
$$CM_SYS_PROP_NS	Properties page
$$CM_TOC_NS	Left-hand frame
$$CM_USR_PROP_NS	Custom properties page (posts data from it)
$$CM_VER_DIFF_NS	Compare versions page
$$CM_VER_INFO_NS	Version history page
$$CM_VER_LINKS_NS	Check links page
$$CM_WEBPUB_NS	Web Publisher applet

Conditional Variables

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:

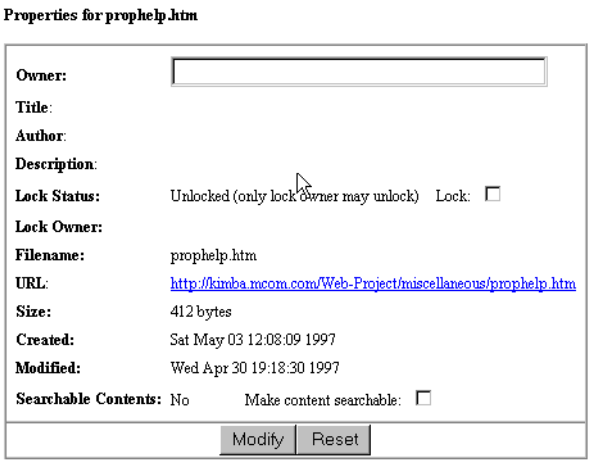
```
variableName[conditionalized output]
```

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>]
```

Figure 15.1 shows the file properties page displayed by the `sys-prop.pat` pattern file. The fields for Owner, Title, Author, Lock Status, URL, and so on are all defined in the pattern file. Most of the variables are in the `dblist.ini` file, but there are a few that are defined by Web Publisher.

Figure 15.1 Figure 1: The Web Publisher file properties page



To see how these work together, here are some of the more interesting lines from the file properties pattern file, `sys-prop.pat`, that define various fields and their labels:

```
<TD><B>Owner:</B></TD>
<TD><input name="CM_OWNER" value="$$CM_OWNER" size="40"</TD>
<TR VALIGN=BASELINE><TD NOWRAP><B>Title</B>:</TD><TD>$$Title</TD></TR>
<TR VALIGN=BASELINE><TD NOWRAP><B>Author</B>:</TD><TD>$$CM_AUTHOR</TD></TR>
<TR VALIGN=BASELINE><TD><B>Lock Status:</B></TD>
<TD>$$CM_LOCK_STAT (only lock owner may unlock) <SPACER type=horizontal size=10></SPACER>
$$CM_LOCK_VAL:<SPACER type=horizontal size=5></SPACER>
<INPUT TYPE="checkbox" NAME="CM_LOCK_STAT" ><BR></TD></TR>
$$IF_DOC_HAS_RENDITON[
<TD NOWRAP><B>Rendition</B>:</TD>
<TD><A HREF="$$CM_URL?$$CM_HTML_REND_NS">HTML</A></TD>
</TR>
]
```

Notice the following aspects of the lines:

- The **owner** field is limited to 40 characters.
- The **title** and **author** fields are read-only.
- The **lock status** information is a checkbox with its own associated label that varies depending on whether the file is already locked or unlocked, as indicated by the value of $$CM_LOCK_VAL.
- The **Rendition** field only appears for files that have renditions available. It includes a pattern variable that points to the HTML version of the file and shows it in the right frame.

Chapter

16

Using Search

The iPlanet Web Server search function allows you to search the contents and attributes of documents on the server. As the server administrator, you can create a customized text search interface tailored to your user community.

Note The Search function is not available on Linux platforms.

This chapter contains the following sections:

- About Search
- Configuring Text Search
- Indexing Your Documents
- Performing a Search: The Basics
- Using the Query Operators
- Customizing the Search Interface

About Search

Server documents can be in a variety of formats, such as HTML, Microsoft Excel, Adobe PDF, and WordPerfect provided that there is a conversion filter available for a particular file format. With the filters, 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. For more information, see "About Collections" on page 412.

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 windows 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.

Note Search cannot work if the Web Publishing collection does not yet exist or has been deleted. If search does not work, restart the server with the web publishing function turned on (the default), and try searching again.

If Search is turned on before Web Publishing then the default collection is not created until after a force index is performed. This happens only if Web Publishing is enabled after Search. The reason that the Web Publishing collection does not show up in search is that at the time the search init is run, the collection has not been created. If you restart the server, then it will show up correctly.

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 are as follows:

- 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, are as follows:

- 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

This section includes the following topics:

- Controlling Search Access
- Mapping URLs
- Deciding Which Words Not to Search
- Turning Search On or Off
- Configuring the Search Parameters
- Configuring Your Pattern Files
- Configuring Files Manually

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 list (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. For more information about setting access control, see Chapter 14, "Controlling Access to Your Server".

You can set your server to check access permissions before displaying search results (by choosing Search and clicking the Search link) as described in "Configuring the Search Parameters" on page 405. When this option 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 (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 13, "Managing Server Content."

For information on how to add a doc root for software virtual servers, see "Adding a Doc Root for Software Virtual Servers," on page 332 in Chapter 13, "Managing Server Content."

Note By default, URLs that are redirected are always escaped. To prevent this, add `escape="no"`. For example:

```
NameTrans fn="redirect" from="/foobar" url-
prefix="index.html" escape="no"
```

The iPlanet Web Server provides five default mappings:

- **`/`**—the primary document directory (sometimes called the **document root**), which initially maps to *server_root*`/docs`
- **`/help`**—the directory for most of the help files

- **`/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 iPlanet Web Server

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.

   The web server displays the Additional Document Directories window.
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/server4/docs/marketing/bizplans
   ```
5. If you want to apply a style to the directory, select the style in the **Apply Style** drop-down list.

   For more information about styles, see Chapter 12, "Working With Configuration 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

You can specify words the search engine should not index or search against. These words are sometimes referred to as stop words or drop words and typically include articles, conjunctions, and prepositions such as at, and, be, for, and the.

To specify stop words, you need to edit the file named style.stp. This file resides in each of the subdirectories `html`, `pdf`, `mail`, and `news` (for each collection type) in the directory *`server_root`*`\plugins\search\common\style`. Each style`.stp` file controls stop words for that collection type; for example, the style`.stp` file in *`server_root`*`\plugins\search\common\style\html` controls stop words for html files in the collection.

Add the stop words to style.stp, one per line and left justified. You can use operators such as square brackets ([]) to indicate character classes, periods (.) to indicate any character, and plus notation (+) to indicate repeats. For example, the style.stp file might contain the following lines:

```
.........................................+
at
and
be
[0-9a-zA-Z]
[0-9][0-9][0-9][0-9]+
```

In this example, the first line of periods (in the file by default) indicates that words with 40 or more characters are not to be indexed as well as the words at, and, and be. [0-9a-zA-Z] indicates that all one letter words are not to be indexed. [0-9][0-9][0-9][0-9]+ indicates that all integers with 4 or more digits are not to be indexed.

The words you specify are case sensitive so if you want to stop all the case variations of a word you need to enter them all. For instance, for the you might enter the, THE, and The.

Make sure you have the stop list you want before you create a collection. If you need to change the stop list after a collection has been created, you need to delete the collection, change the stop list for the collection type, recreate the collection, and reindex all the documents in the collection.

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 search off, the search plug-in is not loaded when the HTTP server starts up. The default is for search to be turned off.

Note If search is turned off, the Find Broken Links function in Web Publisher is not available because it executes a search as part of its operation.

To turn off the search function, use The Search State Page in the Server Manager.

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 **Search**.
2. Click the **Search Configuration** link.

   The web server displays the Search Configuration window.
3. Type the default maximum number of search result items displayed to users at a time.

   This number 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 16.1.

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 HTML 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 on a collection before displaying the search results, click **Yes** under the label **Check access permissions on collection root before doing a search?**

   If you click Yes, the server checks the user's access privileges for each collection before displaying the documents found as a result of the search. Only the documents in a collection that you have permission to view are displayed.

8. Click OK to set your new search configuration.

Table 16.1 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

Table 16.1 Common Posix date and time formats

Format	Displayed result (example)
%y	Year without century (for example, 00-99)
%Y	Year with century (for example, 1999)

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. For more information about how to change the user interface, see "Customizing the Search Interface" on page 443.

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.

To configure pattern files, perform the following steps:

1. From the Server Manager, choose **Search**.
2. Click the **Search Pattern Files** link.

   The web server displays the Search Pattern Files window.
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 Files 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 iPlanet Web Server's Search pages, 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" on page 443.

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.

This section includes the following topics:

- The Configuration Files
- Adjusting the Maximum Number of Attributes
- Restricting Memory for Indexing
- Restricting Your Index File Size
- Removing Access to the Web Publishing Collection

The Configuration Files

The configuration files that govern searching are described in the following list:

- **`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 windows, 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 iPlanet Web Server's windows.

- **`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. For more information, see "User-defined Pattern Variables" on page 449.

- **`dblist.ini`**—This collection contents file describes collection-specific information. 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. For more information about the attributes for each format, see "About Collection Attributes" on page 413 and Table 16.2.

You can use the Add Custom Property window to add additional properties for the Web Publishing collection. 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 Maintain Collection window (choose Search and click the Maintain Collection link) to remove the collection and then use the New Collection window (click the New Collection link) 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 server is installed on a machine that has less than the suggested minimum RAM requirement.
- 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.

Removing Access to the Web Publishing Collection

Web Publishing appears in the Search In field of the user's standard search query page. To remove the Web Publishing collection from this field, you need to edit the `dblist.ini` file as follows:

1. In the "`[web_htm]`" section, change "`NS-display-select=YES`" to "`NS-display-select=NO`".
2. Restart the server.

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.

Note Search cannot work if the Web Publishing collection does not yet exist or has been deleted. If search does not work, restart the server with the web publishing function turned on (the default), and try searching again.

This section includes the following topics:

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 "Displaying Collection Contents" on page 432.

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 multi-format collections are converted to HTML, if you have keyview filters installed. For more information, see:

```
http://www.keyview.com)
```

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.

When you have multiple server instances, the collection you create is only associated with the server instance on which the collection was created. Therefore, users can only search collections for that server instance.

About Collection Attributes

Certain file formats have a default set of attributes that are indexed for files of that type, as shown in Table 16.2.

Table 16.2 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. Used by the web publishing and other multi-format collections.
NEWS	`From`	text	The source userID of the news item.
	`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.

Table 16.2 The default attributes indexed for each file format

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" on page 409.

For example, a document could have these lines of HTML code:

```
<META NAME="Writer" CONTENT="R. Hunter">

<META NAME="Song" CONTENT="Stella Blue">
```

If this document was indexed with its META tags extracted, you could search it for specific values in the writer or product fields. For example, you could enter this query: `Writer <contains> Hunter` or `Song <contains> Blue`.

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. You can use the Add Custom Property window (choose Web Publishing and click the Add Custom Property link) to redefine the text-formatted dates and numbers so that you can perform searches based on actual dates and numbers for data in the Web Publishing collection.

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.

You can only have 12 collections on your server, which is limited to 10 user-defined collections for any server that uses web publishing. If you want to use a 13th collection, you must remove one of your existing collections (choose Search and click the Maintain Collection link). Do not remove the web publishing collection if one exists for your server.

You can only have entries for a maximum of 16 million documents in your collections. A document that is indexed in multiple collections counts as multiple documents. It is best to create new collections of over 10,000 documents at low-traffic times, or the indexing operation may affect your system's performance.

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" on page 411.

To create a new collection, perform the following steps:

1. From the Server Manager, choose **Search**.
2. Click the **New Collection** link.

   The web server displays the Create a Collection window. 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. For more information about additional document directories, see "Mapping URLs" on page 402.
3. You can select any of the items in the drop-down list as a starting point for finding the directory you want to index.
4. If you want to index a different subdirectory, click the **View** button to see a list of resources.
5. 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 window and the directory name appears in the **Directory to Index** field.

6. You can index all HTML 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. For details of the syntax for wildcard patterns, see "Using Wildcards" on page 440.

Note You cannot index a file that includes a semi-colon (;) in its name. You must rename such files before you can index them.

7. To index the subdirectories within the specified directory, click **Include Subdirectories**.

8. Type a name for your collection in the **Collection Name** field.

   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.

Note Do not use accented characters in the collection name. If you need accented characters, exclude the accents from the collection name, but use accented characters in the label. The label is what is displayed to the user from the search interface.

9. Type a user-defined name for your collection in the optional **Collection Label** field.

   This name 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.

10. Type a description for your collection (up to a maximum of 1024 characters) in the optional **Description** field.

    This description is displayed in the collection contents page.

11. Select the type of files the collection is to contain: ASCII, HTML, news, email, or PDF.

    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. For information about the attributes for each format, see Table 16.2 and "About Collection Attributes" on page 413.

    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.

    Complex PDF files, such as those that are password protected or that contain graphical navigation elements cannot be correctly converted when they are indexed as part of a multi-format collection. The file data converts correctly when they are part of a PDF-only collection. Graphic elements are not converted.

12. 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 thirty (30) different user-defined META tags in a document. You can only use this option for HTML collections.

13. 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 13, "Managing Server Content."

14. Click OK to create a new collection.

Note Once you begin indexing a collection, you cannot stop the process until either the indexing is complete or you reboot the system. Shutting down your server does not kill the process.

Configuring a 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. For more information about the configuration files, see "Configuring Files Manually" on page 408. 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 window 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 changes to this collection's settings.

To configure a collection, perform the following steps:

1. From the Server Manager, choose **Search**.
2. Click the **Configure Collection** link.

   The web server displays the Configure Collection window.
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.
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. For more information about additional document directories, see "Mapping URLs" on page 402.

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 a 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, perform the following steps:

1. From the Server Manager, choose **Search**.
2. Click the **Update Collection** link.

   The web server displays the Update Collection window.
3. Select the collection you want to update from the drop-down list.

   The 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 a 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 not done automatically, so you must manually optimize after you reindex or update a collection. 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.

Note Do not use your local file manager to remove collections, especially not the web publishing collections. If by chance you do, when you try to execute a search before restarting your server again, the search will fail even if it doesn't use the web publishing collection. Once you restart your server, a new web publishing collection will be automatically created for you, so your search can execute.

To perform any of the collection management tasks, use The Maintain Collection Page in the Server Manager.

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.

A common combination of tasks is to set up a pair of regularly scheduled reindex and update operations to clean out deleted entries an to add entries for new documents matching your collection criteria.

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 not done automatically, so you must manually optimize after you reindex or update a collection. 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 add entries for new documents but cleans up the collection by removing entries to files that have been deleted.

You can update a collection, by entering new indexing criteria for the collection, say `*.html`, which adds any new documents that match the criteria.

To optimize, reindex, or update your collection, perform the following steps:

1. From the Server Manager, choose **Search**.
2. Click the **Schedule Collection Maintenance** link.

   The web server displays the Schedule Collection Maintenance window.
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, Optimize, or Update.

You can set up different schedules for different operations on the same collection.

5. If you choose to update your collection, two extra fields are displayed for entering the document matching criteria and for including documents found in subdirectories that match your criteria.

6. 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.

7. 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.

8. Click OK to schedule the maintenance.

For Unix/Linux users, to make your newly scheduled maintenance take effect, you must restart the `ns-cron` process from the Administration Server.

To restart the `ns-cron` process, perform the following steps:

1. From the Administration Server, Choose **Global Settings**.

2. Click the **Cron Control** link.

3. If `ns-cron` is already on, click **Restart** to restart it. If `ns-cron` is not on, click **Start** to start it up.

   In either case, your regularly scheduled maintenance will now be able to take place.

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 unschedule collection maintenance, perform the following steps:

1. From the Server Manager, choose **Search**.
2. Click the **Remove Scheduled Collection Maintenance** link.

   The web server displays the Remove Scheduled Collection Maintenance window.
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.

For Unix/Linux users, to make your newly scheduled maintenance take effect, you must restart the `ns-cron` process.

To restart the `ns-cron` process, perform the following steps:

1. From the Administration Server, choose **Global Settings**.
2. Click the **Cron Control** link.
3. If `ns-cron` is already on, click **Restart** to restart it. If `ns-cron` is not on, click **Start** to start it up.

   In either case, your regularly scheduled maintenance will no longer take place.

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 iPlanet Web 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.

Note If the search function is turned off, these query forms are not available.

This section includes the following topics:

- Search Home Page
- A Search Query
- Guided Search
- Advanced Search
- The Search Results
- Displaying Collection Contents

Search Home Page

The search home page (see: `http://`*`serverid`*`:port/search`) 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 iPlanet Web 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).

Note You can only execute date and number comparison searches against HTML META attribute values in the web publishing collection provided you have redefined them as date or number properties through the Web Publishing | Add Custom Property form.

To perform a standard search, perform the following steps:

1. Type the following URL in the location field in your web browser:

   `http://serverid:port/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. For details about the search operators, see "Using the Query Operators" on page 432.

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.

Note The attributes for Version Control and Link Management are no longer used in iPlanet Web Server. However, note that if you perform a guided search, iPlanet Web Server may still return them; consequently, do not use these variables.

There are two ways to obtain the guided search page: through the Search home page or through the standard search query page.

To access the guided search interface through the Search home page, perform the following steps:

1. Type the following URL in the location field in your web browser:

   `http://serverid:port/search`

2. Click the **Guided Search** link on the home page.

To access guided search through the standard search query page, perform the following steps:

1. Go to the standard search query page by typing the following URL in the **location** field in your web browser:

   `http://serverid:port/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. For details about the search operator, see “Using the Query Operators” on page 432.

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, perform the following steps:

1. Type the following URL in the location field in your web browser:

   `http://serverid:port/search`

2. Click the **Advanced HTML Search** link on the home page.

To access advanced HTML search through the standard search query page, perform the following steps:

1. Go to the standard search query page by typing the following URL in the **location** field in your web browser:

   `http://serverid:port/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 web server displays the advanced HTML query page.

4. In the **For** field, type in the word or phrase you want to search for. You can create complex queries by combining operators. For details about the search operators, see "Using the Query Operators" on page 432.

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. For more information about sorting, see "Sorting the Results" on page 431.

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, but the web publishing collection cannot be used in a multi-collection search.

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.

Your access permissions are checked at several points during the search process:

- When a user clicks on the icon displayed for a document in the search results, which displays the highlighted version.
- When searching on a collection other than Web Publishing that has the option NS-collection-acl-check set to yes. (NS-collection-acl-check is set in the `webpub.conf` file and applies to all collections. When it is set, ACLs that are set on URIs matching the primary document directory defined for the collections (in `dblist.ini`) will be honored by not allowing search to be done on those collections.)
- Whenever a user searches on the Web Publishing collection.

Listing Matched Documents

In the default installation of the iPlanet Web 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. For information about the attributes for each format, see "About Collection Attributes" on page 413.

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 Add Custom Property page from the Web Publishing tab and check the box that marks this property as a META-tagged attribute.

Displaying a Highlighted Document

In the default installation of iPlanet Web 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 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 icon shown next to the document's listing. When you click it, there is additional code defined behind the icon's link to format the displayed document with the search query highlighted.

In the default search results page, if you click the file's URL you open 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 iPlanet Web 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, use the following URL:

```
http://serverid:port/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. Only those queries that can have varying degrees of matching can be ranked. For example, <CONTAINS queries either do or do not contain the given string, but <NEAR queries can be ranked according to how close the words are to each other: words closer together are listed at the top of the search results, while those that are far apart are put at the bottom of the results.

This section includes the following topics:

- Default Assumptions
- Search Rules
- Determining Which Operators To Use
- Using Wildcards

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 i*s 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 that 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 required 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 16.3 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*.

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 16.4 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 16.4 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 16.4 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 16.4 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 16.5 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.

^	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 the following string:

```
<WILDCARDZine\\*\\*\\*
```

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 the following string:

```
<WILDCARD‘a{b‘
```

For another example, if you wanted to search on the string “c`t”, which contains a back quote, you would type the following string:

```
<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 Files Manually" on page 408).

Note The search home page, at `http://serverid:port/search` 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.

This section includes the following topics:

- Dynamically Generated Headers and Footers
- HTML Pattern Files
- Search Function Syntax
- Using Pattern Variables

Dynamically Generated Headers and Footers

You can specify dynamically generated headers and footers. To accomplish this, add the `add-headers` and `add-footers` directives to your `obj.conf` file as Service functions. These directives take either a path or `uri` parameter. Use the path parameter to specify a static file as the header or footer. For example:

```
Service fn="add-headers" path="/export2/docs/header.html"
Service fn="add-footer" path="/export2/docs/footer.html"
```

Use the `uri` parameter to specify a dynamically generated file, such as a CGI program, as the header or footer. For example:

```
uri="/cgi-bin/header.cgi"
```

These Service functions should precede the actual Service function that will answer the request, such as `send-file` or `send-cgi`.

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 iPlanet Web 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. A file prefix designates which type of file the pattern file is for, for example, ASCII-record.pat, EMAIL-record.pat, etc. The following list describes the general pattern file types:

- **`NS-query.pat`** displays the standard and advanced query pages. Contains HTML calling the Web Search (the "Search the Web" box) as part of the search query page.
- **`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 (for more information, see "Displaying a Highlighted Document" on page 431).
- **`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" on page 448):

- user defined, in the `userdefs.ini` file, with a `$$` prefix (see "User-defined Pattern Variables" on page 449).
- defined in the configuration files, `webpub.conf` and `dblist.ini` files, with a `$$NS-` prefix (for more information, see "Configuration File Variables" on page 451).
- search macros and variables generated by a pattern file, with a `$$NS-` prefix (for more information, see "Macros and Generated Pattern Variables" on page 453).

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 Files Manually" on page 408.

- **`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://serverid/
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:

```
$$Title[<PTitle: <B$$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. The following table shows the most commonly used URL codes.

Table 16.6 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 iPlanet Web 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]
NS-character-set=iso-8859-1
uidir =  $$NS-server-url/search-ui
icondir = $$uidir/icons
l10nicondir = $$uidir/icons
htmldir = $$uidir/text
logo = <img src="$$icondir/magnifier.jpg" border=0
align=absmiddle><b><font size=+2>N</font><font size=+1>etscape </
font><font size=+2>S</font><font size=+1>earch</font></b>
sitename = $$NS-host
help = /help/5search.htm
title = Sample Search Interface
```

```
searchButtonLabel = Search
searchNote = To search, choose a collection, then enter words and
phrases, separated by commas<br>(e.g., search, jet engines, basketball).

advSearchNote = To search, choose collections, then enter words and
phrases, separated by commas<br>(e.g., search, jet engines,
basketball).<p>Sorting is done on any defined attributes. Use '-' to
specify descending order sort<br>(e.g., Title,-Author,+Date)
queryLabel = For:
queryLabelSJIS = $$queryLabel
queryLabelEUC = $$queryLabel
queryLabelJIS7 = $$queryLabel
collectionLabel = Search in:
booleanLabel = Boolean
sortByLabel = Sort by:
sortByLabelSJIS = $sortByLabel
sortByLabelEUC = $sortByLabel
sortByLabelJIS7 = $sortByLabel
freetextLabel = Freetext (unavailable)
maxDocumentsLabel = Documents to return:
maxDocumentsLabelSJIS = $$maxDocumentsLabel
maxDocumentsLabelEUC = $$maxDocumentsLabel
maxDocumentsLabelJIS7 = $$maxDocumentsLabel
copyright = Copyright &#169; 1997 Netscape Communications Corporation.
All Rights Reserved.
advancedButtonLabel = Advanced Button Label
helpButtonLabel = Help Button Label
```

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 "Macros and Generated Pattern Variables" on page 453.

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 the following table:

Table 16.7 Common HTML character entities

Numeric code	Entity name	Description
&#032;		Space
&#034;	&quot;	Quotation mark
&#036;	$	Dollar sign
&#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 iPlanet Web 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 the following table:

Table 16.8 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 iPlanet Web Server includes variables for the web publishing collection. Among the variables defined there are:

```
NS-collection-alias = Web Publishing
NS-doc-root = C:/Netscape/server4/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 11.9 contains some of the more commonly found collection-specific variables.

Table 16.9 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 the following table:

Table 16.10 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.

Table 16.10 Macros and generated pattern variables

Variable	Description
`$$NS-docs-found`	The actual number of documents that the search engine found for this request.
`$$NS-docs-matched`	The number of documents returned from the search (up to `NS-max-records`) for this request.
`$$NS-docs-searched`	The number of documents searched through for this request.
`$$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.

Appendixes

- **HyperText Transfer Protocol**
- **ACL File Syntax**
- **Internationalized iPlanet Web Server**
- **Server Extensions for Microsoft FrontPage**

Appendix

A

HyperText Transfer Protocol

This appendix provides a short introduction to a few HyperText Transfer Protocol (HTTP) basics. For more information on HTTP, see the Internet Engineering Task Force (IETF) home page at:

```
http://www.ietf.org/home.html
```

This appendix contains the following sections:

- About HyperText Transfer Protocol (HTTP)
- Requests
- Responses

About HyperText Transfer Protocol (HTTP)

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 iPlanet Web Server 4.x 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
```

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 16.11.

Table 16.11 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.
`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 16.12 contains some common status codes.

Table 16.12 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.

Table 16.12 Common HTTP status codes

Status code	Meaning
`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 16.13.

Table 16.13 Common response headers

Response header	Description
`Server`	The name and version of the web server.
`Date`	The current date (in Greenwich Mean Time).
`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

ACL File Syntax

This appendix describes the access-control list (ACL) files and their syntax. ACL files are text files that contain lists that define who can access resources stored on your web server. By default, the web server uses one ACL file that contains all of the lists for access to your server. However, you can create multiple ACL files and reference them in the `obj.conf` file.

You need to know the syntax and function of ACL files if you plan on customizing access control using the access-control API. For example, you might use the access control API to interface with another database, such as an Oracle or Informix database. For more information on the API, see the iPlanet documentation site at:

```
http://www.iplanet.com/docs
```

This appendix contains the following sections:

- ACL File Syntax
- Referencing ACL Files in obj.conf

ACL File Syntax

All ACL files must follow a specific format and syntax. An ACL file is a text file containing one or more ACLs. All ACL files must begin with the version number they use. There can be only one version line and it can appear after any comment lines. For example:

```
version 3.0;
```

You can include comments in the file by beginning the comment line with the # sign.

Each ACL in the file begins with a statement that defines its type. ACLs can follow one of three types:

- **Path ACLs** specify an absolute path to the resource they affect
- **URI (Uniform Resource Indicator) ACLs** specify a directory or file relative to the server's document root.
- **Named ACLs** specify a name that is referenced in resources in the `obj.conf` file. The server comes with a "default" named resource that allows read access to anyone and write access to users in the LDAP directory. Even though you can create a named ACL from the iPlanet Web Server windows, you must manually reference the named ACLs with resources in the `obj.conf` file.

The type line begins with the letters `acl` and then includes the type information in double-quotation marks followed by a semicolon. Each type information for all ACLs must be a unique name--even among different ACL files. The following lines are examples of several different types of ACLs:

```
acl "path=C:/Netscape/server4/docs/mydocs/";
acl "*.html";
acl "default";
acl "uri=/mydocs/";
```

After you define the type of ACL, you can have one or more statements that define the method used with the ACL (authentication statements) and the people and computers who are allowed or denied access (authorization statements). The following sections describe the syntax for these statements.

Authentication Statements

ACLs can optionally specify the authentication method the server must use when processing the ACL. There are two general methods:

- Basic requires users to enter a username and password before accessing a resource.
- SSL requires the user to have a client certificate. For this method to work, the web server must have encryption turned on, and the CA must be in the list of trusted CAs.

By default, the server uses the Basic method for any ACL that doesn't specify a method.

Each authenticate line must specify what list (users, groups or both) the server should use when authenticating users. The following authentication statement, which would appear after the ACL type line, specifies basic authentication with users matched to individual users in the database or directory:

```
authenticate (user) {
     method = “basic”;
};
```

The following example uses SSL as the authentication method for users and groups:

```
authenticate (user, group) {
     method = “ssl”;
};
```

The following example allows any user whose username begins with the letters `sales`:

```
authenticate (user)
allow (all)
    user = sales*
```

If the last line was changed to `group = sales`, then the ACL would fail because there are no groups in the user lists.

Authorization Statements

Each ACL entry can include one or more authorization statements. Authorization statements specify who is allowed or denied access to a server resource. Use the following syntax when writing authorization statements:

```
allow|deny [absolute] (right[,right...]) attribute
expression;
```

Start each line with either allow or deny. It's usually a good idea to deny access to everyone in the first rule and then specifically allow access for users, groups, or computers in subsequent rules. This is because of the hierarchy of rules. That is, if you allow anyone access to a directory called `/my_stuff`, and then you have a subdirectory `/my_stuff/personal` that allows access to a few users, the access control on the subdirectory won't work because anyone allowed access to the `/my_stuff` directory will also be allowed access to the `/my_stuff/personal` directory. To prevent this, create a rule for the subdirectory that first denies access to anyone and then allows it for the few users who need access.

However, in some cases if you set the default ACL to deny access to everyone, then your other ACL rules don't need a "deny all" rule.

The following line denies access to everyone:

```
deny (all)
    user = "anyone";
```

Hierarchy of Authorization Statements

ACLs have a hierarchy that depends on the resource. For example, if the server receives a request for the document (URI) `/my_stuff/web/presentation.html`, the server first looks for an ACL that matches the file type or any other wildcard pattern that matches the request, then it looks for one on the directory, and finally it looks for an ACL on the URI. If there are more than one ACLs that match, the server uses the last statement that matches. However, if you use an absolute statement, then the server stops looking for other matches and uses the ACL containing the absolute statement. If you have two absolute statements for the same resource, the server uses the first one in the file and stops looking for other resources that match.

For example, using the ACL hierarchy with the request for the document `/my_stuff/web/presentation.html`, you could have an absolute ACL that restricts access to the file type `*.html`. Then the server would use that ACL instead of looking for one that matches the URI or the path.

```
version 3.0;
acl "default";
authenticate (user,group) {
     prompt="WebServer Server";
};
allow (read,execute,list,info)
     user = "anyone";
allow (write,delete)
     user = "all";
acl "*.html";
deny absolute (all)
     user = "anyone";
acl "uri=/my_stuff/web/presentation.html";
deny (all)
     user = "anyone";
allow (all)
     user = "anyone";
```

Attribute Expressions

Attribute expressions define who is allowed or denied access based on their username, group name, host name, or IP address. The following lines are examples of allowing access to different people or computers:

- `user = “anyone”`
- `user = “smith*”`
- `group = “sales”`
- `dns = “*.iplanet.com”`
- `dns = “*.iplanet.com,*.mozilla.com”`
- `ip = “198.*”`
- `ciphers = “rc4”`

You can also restrict access to your server by time of day (based on the local time on the server) by using the `timeofday` attribute. For example, you can use the `timeofday` attribute to restrict access to certain users during specific hours.

Note Use 24-hour time to specify times (for example, use 0400 to specify 4 a.m. or 2230 for 10:30 p.m.).

The following example restricts access to a group of users called guests between 8 a.m. and 4:59 pm.

```
allow (read)
     (group="guests") and
     (timeofday<800 or timeofday=1700);
```

You can also restrict access by day of the week. Use the following three-letter abbreviations to specify days of the week: Sun, Mon, Tue, Wed Thu, Fri, and Sat.

The following statement allows access for users in the premium group any day and any time. Users in the discount group get access all day on weekends and on weekdays anytime except 8am-4:59pm.

```
allow (read) (group="discount" and
dayofweek="Sat,Sun") or
(group="discount" and (dayofweek="mon,tue,wed,thu,fri"
and
(timeofday<0800 or timeofday=1700)))
or
(group="premium");
```

Operators For Expressions

You can use various operators in attribute expressions. You can use parentheses to delineate the order of precedence of the operators. With `user`, `group`, `dns`, and `ip`, you can use the following operators:

- and
- `or`
- not
- `=` (equals)
- `!=` (not equal to)

With `timeofday` and `dayofweek`, you can use the following additional operators:

- greater than
- `<` less than

- = greater than or equal to
- <= less than or equal to

The Default ACL File

After installing the server, the server uses the default settings in the file `server_root/httpacl/generated.https-serverid.acl`. There is also a file called `genwork.https-serverid.acl` that is a working copy the server uses until you save and apply your changes when working with the user interface. When editing the ACL file, you might want to work in the `genwork` file and then use the iPlanet Web Server to save and apply the changes.

The following text is from the default file:

```
# File automatically written
#
# You may edit this file by hand
#
version 3.0;
acl "agents";
authenticate (user,group) {
          prompt = "WebServer Server";
};
deny (all)
    user = "anyone"
allow absolute (all)
    user = "all";
acl "default";
allow (read,execute,list,info)
    user = "anyone";
allow (write,delete)
    user = "all";
```

The default ACL file is referenced in `magnus.conf` as follows:

```
ACLFile absolutepath/generated.https-serverid.acl
```

You can reference multiple ACL files in magnus.conf and then use their ACLs for resources in `obj.conf`. However, the server uses only the first ACL file with the web publisher and when evaluating access control for objects that don't have specific ACLs listed in `obj.conf`. If you're using the iPlanet Web

Server windows to do some access control, the first ACL file in magnus.conf should point to the file `generated.https-serverid.acl`. See the section "Referencing ACL Files in obj.conf" on page 472 for more information.

General Syntax Items

Input strings can contain the following characters:

- Letters a through z
- Numbers 0 through 9
- Period and underscore

If you use any other characters, you need to use double-quotation marks around the characters.

A single statement can be placed on its own line and be terminated with a semicolon. Multiple statements are placed within braces. A list of items must be separated by commas and enclosed in double-quotation marks.

Referencing ACL Files in obj.conf

If you have named ACLs or separate ACL files, you can reference them in the `obj.conf` file. You do this in the `PathCheck` directive using the `check-acl` function. The line has the following syntax:

```
PathCheck fn="check-acl" acl="aclname"
```

The `aclname` is a unique name of an ACL as it appears in any ACL file.

For example, you might add the following lines to your `obj.conf` file if you want to restrict access to a directory using the acl named `testacl`:

```
<Object ppath="/usr/ns-home/docs/test/*"
PathCheck fn="check-acl" acl="testacl"
</Object
```

In the previous example, the first line is the object that states which server resource you want to restrict access to. The second line is the `PathCheck` directive that uses the `check-acl` function to bind the name ACL (`testacl`) to the object in which the directive appears. The `testacl` ACL can appear in any ACL file referenced in `magnus.conf`.

Appendix

C

Internationalized iPlanet Web Server

The internationalized version of the iPlanet Web 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.

This appendix contains the following sections:

- General Information
- Server-side JavaScript Information
- Search Information
- Getting Support for Accented Characters in Filenames

General Information

The following information covers the international considerations for general server capabilities.

This section includes the following topics:

- Installing the Server
- Entering 8-bit Text

- Using the Accept Language Header
- Language Settings in Configuration Files

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 iPlanet Web Server, Enterprise Edition 4.1 *Release Notes*. You can access the *Release Notes* online via the link provided in 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 4.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.

Note iPlanet Web Server 4.x no longer packages the `ldapsearch` and `ldapmodify` utilities. Earlier versions of Enterprise Server included them, since those versions employed local LDAP database support. iPlanet Web Server 4.x now uses an LDAP server all the time which includes these utilities.

Note The default maximum number of parallel LDAP sessions is now set to 8. There is a way to override this limitation. In addition to the `binddn` and `bindpw` properties that a LDAP connection listed in `dbswitch.conf` may have, iPlanet Web Server 4.x now includes a `sessions` property. The value is numeric and this property sets the maximum number of parallel connections in the LDAP session pool.

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`

Language Settings in Configuration Files

The following directives in the `magnus.conf` file affect languages:

Table 16.14 International settings in magnus.conf

Directive	Values	Description
`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`.
`DefaultLanguage`	`en`, `fr`, `de`, `ja`	Specifies the language used if a resource cannot be found for the client language or the administration language.
`AcceptLanguage`	`on`, `off`	Enables or disables the Accept language header parsing.

The following directives in the `ns-admin.conf` file affect languages:

Table 16.15 International settings in ns-admin.conf

Directive	Values	Description
`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`.
`AdminLanguage`	`en`, `fr`, `de`, `ja`	Sets the language used for administrative pages that are accessed through the administration server.
`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 16.16 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 16.17 Options for the jsac compiler

Option	Usage
`-c`	Check only; do not generate *`binaryFile`*
`-v`	Enable verbose output
`-d`	Enable debug output
`-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 16.18 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, you need to install Oracle and set up your environment, verify the connection, and verify the language setup. follow these overall steps. This section discusses these topics:

- Installing Oracle and Setting Up Your Environment
- Verifying the Connection
- Verifying 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. Note that the environment variable syntax assumes C Shell.

Server-side JavaScript library:

- `setenv LD_LIBRARY_PATH` *`server_root`*`/bin/https:$LD_LIBRARY_PATH`

Environment variables for Oracle:

- `setenv ORACLE_HOME` *`oracle_root`*
  for example, `/usr/oracle7`
- `setenv ORACLE_SID` *`oracle_service_ID`*
  for example, `WG73`
- `setenv TNS_ADMIN` *`path_to_tnsnames.ora`*
  `for example, /.../tnsnames.ora`

Environment variable for NLS (National Language Support) in Oracle:

- `setenv NLS_LANG` *`language_charset_info`*
  for example, `japanese_japan.JA16EUC`

  (This example sets up `x-euc-jp`)

Environment variable for the path:

- `setenv PATH` *`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.

Table 16.19

Field	Value
Server Type	`ORACLE`
Server Identifier	`WG73`
User ID	`system`

Table 16.19

Field	Value
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:

   ```
   setenv TWO_TASK SERVER SID alias
   ```

3. Set the `NLS_LANG` environment variable to the correct client language and character set information.
4. 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)
   ```

5. Click **Add New Customer** and enter data in the character set you specified.
6. Click **Home** to go back to the `videoapp` home page, and then click **Save Changes**.
7. Click **Delete a Customer**.
8. 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. For best results, use 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 16.20 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
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.

Getting Support for Accented Characters in Filenames

If the filenames on your server contain accented characters, for instance elninõ.html, you can get support for them by specifying the 8859 character set as the internal coding for search collections. To specify 8859, you need to modify the file `language.conf` in the directory `<serverRoot>\plugins\search\admin`. This file is used by the Search Engine, Web Publishing, and the document indexing features of the server.

The `language.conf` file contains the following lines for the English language. These lines direct the server to configuration files that use 8859 as the default character set. The configuration files are located in the directory `<serverRoot>\plugins\search\common`.

```
# [en-ns]
# name = English NS 8859 (ISO-8859-1)
# lang = english-ns;8859
# charset = iso-8859-1
# cjk = N
# encode850 = N
```

To specify 8859, you need to activate these lines in `language.conf` by removing the comment characters (#).

If you make this change to the `language.conf` file after a collection has been created, to support accented characters in filenames for that collection you need to delete the collection, make this change to the file, recreate the collection choosing "English NS 8859 (ISO-8859-1" from the "Documents are in" drop-down list, and reindex all the documents in the collection.

Appendix

D

Server Extensions for Microsoft FrontPage

This appendix describes using server extensions on your iPlanet Web Server that provide support for Microsoft's FrontPage. These extensions provide the internal server-side support you need if you are using FrontPage webs.

This appendix includes the following sections:

- Overview
- Downloading the Extensions
- Installing FrontPage Server Extensions
- Further Information

Overview

FrontPage server extensions are CGI programs that provide iPlanet Web Server support for FrontPage webs. Client-server communication takes place through standard HTTP POST requests that are forwarded to the appropriate extension's CGI program. If you use FrontPage webs, the extensions provide support for FrontPage authoring and publishing, access permission, and WebBot functions. For example:

- When a user moves a page between folders in a FrontPage web, the extensions automatically update all links to that page from every other page in the web.
- You can specify which users have permission to administer, author or browse a FrontPage web.
- When FrontPage web users participate in a discussion group, the extensions take advantage of the available WebBots to maintain an index of links to discussion articles, tables of contents, and search forms.

The extensions can minimize file transfers over the Internet. For example, when a user opens a FrontPage web from an iPlanet Web Server with the extensions, web metadata, such as its map of links, is downloaded to the user's machine but the full set of web pages remain on the server. A page is downloaded only when it is opened for editing.

Once you have installed the extensions on your server, FrontPage web publishing, administering, and discussion group functionality is available from any computer that is on the Internet or a local Intranet, although you need the FrontPage client program for authoring and administrative functions.

This section includes the following topics:

- Types of FrontPage Webs
- Domain Names and FrontPage Webs
- Security Issues

Types of FrontPage Webs

There are two kinds of FrontPage webs: *root webs* and *sub-webs*. A **root web** is a FrontPage web that is the top-level content directory of a Web server or, in a multi-hosting environment, of a virtual Web server. There can only be one root web per Web Server or virtual Web server.

A single root web can support a number of sub-webs. A **sub-web** is a complete FrontPage web that is a subdirectory of the root web. Sub-webs can only exist one level below the root web. Each sub-web can have many levels of subdirectories, making up its content.

Even though sub-webs appear below the root web in the Web server's file system and URL space, the root web does not include the content in its sub-webs. This separation of content is done by the FrontPage Server Extensions.

The root web and all sub-webs on a server must have separate copies of the extensions installed or have stub executables of the extensions programs. Having separate copies of the extensions for each FrontPage web lets the server administrator enforce different end-user, author, and administrator permissions on each FrontPage web, since FrontPage uses the server's built-in security mechanism to control access.

Domain Names and FrontPage Webs

FrontPage webs can be implemented on an iPlanet Web Server and accessed by web browsers in the following ways:

- As private domain names, such as `www.mycompany.com`. These are usually implemented as virtual servers on the same physical server machine using multi-hosting. Private domain name customers each get their own root web and have the option of creating sub-webs.
- As a common or shared domain but with private virtual servers, as in `www.mycompany.myprovider.com`, where `myprovider.com` is a shared domain and `www.mycompany` is a private virtual server. Private virtual server customers on a shared domain each get their own root web and have the option of creating sub-webs.
- As a URL on an Internet service provider's server machine, as in `www.myprovider.com/mycompany`. URL customers get a single sub-web.

Security Issues

FrontPage implements web security on your web server by changing the access-control lists (ACLs) for all files and directories in each FrontPage web. Installing FrontPage always modifies the ACLs of the Server Extensions stub executables contained in the `/_vti_bin` directory in each web. A new installation of FrontPage will additionally modify the ACLs of the web content files, but an upgrade of an existing installation of the Server Extensions will not

modify the content file ACLs and consequently will leave the security settings at a less secure level than the default FrontPage settings. You can upgrade the ACLs of your web content by using the Check and Fix option of the FrontPage Server Administrator utility.

In addition to modifying the security ACLs of the web content files, FrontPage modifies the ACLs of any system DLLs that are used as a result of a FrontPage DLL call, to ensure that the system DLLs will have the correct level of permissions to run under any administrator, author, or end-user's account. For the complete set of ACLs set on FrontPage files, along with a discussion of security considerations when installing the Server Extensions and the reasons why the ACLs of the system DLLs must be modified, see the additional resources available at Ready-to-Run Software and Microsoft's web sites.

Downloading the Extensions

The first step towards installing the extensions is to download them. You can use Microsoft's FrontPage sites or, if you want to install the Unix/Linux extensions, you can use Ready-to-Run Software's site, which also provides a great deal of information and instruction.

- FrontPage 97 Server Extensions (version 2.0):
  - [NT] You can download an executable file.
  - [Unix/Linux] You can download from Ready-to-Run Software's web site an install script and a set of server extensions. Download two tar files for your platform (for Solaris, they are `vt20.solaris.tar.z` and `wpp.solaris.tar.z`, which is part of the WPP Kit Software).
  - [Unix/Linux] You can download from Microsoft's web site an install script and a set of server extensions. Download two tar files for your platform (for Solaris, they are `vt20.solaris.tar.z` and `wpp.solaris.tar.z`, which is part of the WPP Kit Software.)
- FrontPage 98 Server Extensions (version 3.0):
  - [NT] You can download an executable file.

  - — [Unix/Linux] You can download from Ready-to-Run Software's web site an install script and a set of server extensions. Download the `fp_install.sh` file and the tar file for your platform (for Solaris, it is `fp30.solaris.tar.z`)

  - — [Unix/Linux] You can download from Microsoft's web site an install script and a set of server extensions. Download the fp_install.sh file and the tar file for your platform (for Solaris, it is fp30.solaris.tar.z).

- FrontPage 2000 Server Extensions (version 4.0):

  - — [NT] You can download an executable file, `fp2kserk.exe`, which gives information on how to set up and use a FrontPage-extended web. You can download a set of server extensions from the Microsoft web site, `fpse2k_x86_ENG.exe`.

  - — [Unix/Linux] You can download an install script and a set of server extensions from the Ready-to-Run Software web site. Download the `fp_install.sh` file and the tar file for your platform (for Solaris, it is `fp40.solaris.tar.Z`).

  - — [Unix/Linux] You can download an install script and a set of server extensions from the Microsoft web site. Download the `fp_install.sh` file and the tar file for your platform (for Solaris, it is `fp40.solaris.tar.Z`).

Before you install the FrontPage Server Extensions, you need to be sure you have enough disk space available on your local machine, that you have a document root directory, that you have enabled authentication, and that you are aware of some important post-install issues such as access permissions.

This section includes the following topics:

- Space Requirements
- Preliminary Tasks
- Some Additional Considerations

Space Requirements

On Windows NT systems, you need to have approximately 6MB of disk space available. The downloaded file is 3MB and the installed files total 2.5MB.

On Unix/Linux systems, you should have at least 32MB available on your server. The Unix/Linux FrontPage extensions need 9MB of disk space in the `/usr/local/frontpage` directory. If you install the extensions onto your web content, you need an extra 5MB per virtual host unless your web content is in the same disk partition as `/usr/local/frontpage`.

Preliminary Tasks

You need to have a document root directory for your iPlanet Web Server, which is created when you start up your server for the first time. This means you must start up your server at least once before installing the extensions.

Some Additional Considerations

- Do not remove any of the internal files needed by FrontPage such as the `.nsconfig` file. Doing so disables access control for content upload.
- You cannot set a web to be restricted to valid end-users only. If you set this, you receive a message that says "This server does not support restricting end user access."
- [Unix/Linux only] When you install the stub extensions, you should set the web owner to be the same as the iPlanet Web Server user. This is so that the FrontPage extensions have write permissions to certain directories, namely the `https-`*`instance`*`/config` directory and the doc root. The `fpsrvadm.exe` script, which installs stub extensions to the webs, asks for the web owner.

Installing FrontPage Server Extensions

You can install the FrontPage 97, the FrontPage 98, or FrontPage 2000 extensions on Windows NT or Unix/Linux platforms. This document provides instructions for the following platforms:

- Windows NT systems
- Unix/Linux systems - FrontPage97 extensions
- Unix/Linux systems - FrontPage98 extensions
- Unix/Linux systems - FrontPage2000 extensions

To install FrontPage extensions to a Netscape Enterprise Server version 3.x or iPlanet Web Server version 4.x, you need to do a custom installation, because we have changed our registry keys.

Installing FrontPage Server Extensions on Windows NT Systems

The installation process for the FrontPage97, FrontPage98, and FrontPage 2000 extensions on a Windows NT system is relatively straightforward. You download and run an executable file, which installs several files and folders on your system. The extensions require a specific directory structure, which is discussed later in this section. After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

These installation instructions are for the standalone FrontPage Server Extensions that are in a self-extracting executable that is downloadable from the Microsoft FrontPage web site.

Note You must log into your NT system as Administrator or have administrator permission to install the FrontPage Server Extensions.

To install FrontPage Server Extensions on Windows NT, perform the following steps:

1. Run the server extensions setup program for your language and processor type.

   For example, for English FrontPage98 extensions, it is the `fp98ext_x86_enu.exe` file. The server extensions are copied to the folder `C:\Program Files\MicrosoftFrontPage\Version 3.0`. For English FrontPage2000 extensions, it is the `fpse2k_x86_ENG.exe` file. The server extensions are copied to the folder `C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40`.

2. After installing FrontPage 2000 with the Server Extensions Resource Kit, launch the Server Extensions Administrator under **Start | Programs | Administrative Tools**, right-click on your machine's host name under **Console Root | FrontPage Server Extensions**, and select **New Web**. Follow the steps in the wizard to select your server instance and configure it for FrontPage Server Extensions.

3. Select the virtual servers on which the FrontPage Server Extensions should be installed and click OK.

4. Enter the name of a new FrontPage administrator account and a password.

5. You can add other administrator accounts after installing the server extensions using the Permissions command in the FrontPage Explorer.

Installing the server extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page's contents are parsed to expand FrontPage components (such as Include components and Substitution components), create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation also updates the text indices and recalculates the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

For FrontPage97 extensions, these components are installed in the `C:\Program Files\Microsoft FrontPage` directory, where C is your default hard drive. The components are as follows:

- The FrontPage Server Extension `.dll` and `.exe` files are copied to the `\bin` subdirectory and to the default `\windows\system` directory.
- The ISAPI (`.dll`) or CGI (`.exe`) files used by FrontPage to implement the Server Extensions functionality in the user's webs are copied to the `\isapi` and `\_vti_bin` directories, respectively. They are also copied into the document root of each virtual server on which you are installing the FrontPage extensions.
- The FrontPage Server Administrator (`fpsrvwin.exe`) and a command line version (`fpsrvadm.exe`) are copied to the `\bin` directory. The FrontPage Server Administrator is a tool for installing, updating, verifying, or removing the FrontPage Server Extensions.

For FrontPage98 extensions, these components are installed in the `C:\Program Files\Microsoft FrontPage\version 3.0` directory, where C is your default hard drive:

- The FrontPage Server Extensions `.dll` and `.exe` files are copied to the `\bin` subdirectory and to the default `\windows\system` directory.
- The three ISAPI (`.dll`) or CGI (`.exe`) files used by FrontPage to implement the Server Extensions functionality in the user's webs are copied to the `\isapi` and `\_vti_bin` directories, respectively. They are also copied into the document root of each virtual server on which the FrontPage extensions are installed.
- The FrontPage Server Administrator (`fpsrvwin.exe`) and a command line version (`fpsrvadm.exe`) are copied to the `\bin` directory. The FrontPage Server Administrator is a tool for installing, updating, verifying, or removing the FrontPage Server Extensions.
- The Server Extensions Resource Kit.
- HTML Administration forms, a set of HTML forms for remotely administering the FrontPage Server Extensions via web browsers. Also a command line utility (`fpremadm.exe`) for remote administration of the FrontPage Server Extensions is installed in the `\bin` directory.

For FrontPage2000 extensions, these components are installed in the `C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40` directory, where C is your default hard drive:

- The FrontPage Server Extensions `.dll` and `.exe` files are copied to the `\bin` subdirectory and to the `\WINNT\system32` directory.
- The three ISAPI (`.dll`) or CGI (`.exe`) files used by FrontPage to implement the Server Extensions functionality in the user's webs are copied to the `\isapi` and `\_vti_bin` directories, respectively. They are also copied into the document root of each virtual server on which the FrontPage extensions are installed.
- The FrontPage Server Administrator (`fpsrvwin.exe`) is not used or installed with FrontPage 2000. You must use the Server Extensions Administrator (under Start | Programs | Administrative Tools) or the command line version (`fpsrvadm.exe`), which is copied to the `\bin` subdirectory. The FrontPage Server Administrator is a tool for installing, updating, verifying, or removing the FrontPage Server Extensions.
- The Server Extensions Resource Kit is copied to the `\serk` subdirectory.
- HTML Administration forms, a set of HTML forms for remotely administering the FrontPage Server Extensions via web browsers, are copied to the `\admcgi` and `\admisapi` subdirectories. Also a command line utility (`fpremadm.exe`) for remote administration of the FrontPage Server Extensions is installed in the `\bin` directory.

Installation also modifies or adds the following files and directories:

- Modifies the `magnus.conf` file
- Modifies the server's configuration file (`obj.conf`) to add `ObjectType` directives, marking three of the `\_vti_` directories as containing executables.
- Adds the following seven subdirectories are created under your server's document root:

```
\_private
\_vti_bin (contains shtml.exe)
\_vti_bin\_vti_adm (contains admin.exe)
\_vti_bin\_vti_aut (contains author.exe)
\_vti_cnf
```

```
\_vti_log
\_vti_pvt
\_vti_txt
\images
```

- Creates `.nsconfig` files in the `\_vti_bin`, `\_vti_adm`,`\_vti_aut` directories and the document root directory.

Once you have completed the installation process, you must also perform the following administrative tasks:

- For FrontPage 97 and 98, execute the `fpsrvwin.exe` file (located in the \bin directory of your FrontPage directory) to set the server port, test the extensions, install the extensions to other virtual servers, and update extensions.
- For FrontPage 2000, run the Server Extensions Administrator (under Start | Programs | Administrative Tools) or the command line version (`fpsrvadm.exe`).
- Select the server and web you want to work with:
  - — A remote machine must have the FrontPage97, FrontPage98, or FrontPage 2000 program installed (Macintosh or Windows only). Once the FrontPage program is started, the user is prompted for the name of a server to edit or open.
  - — If the user wants to edit a web on a different machine, click on "MoreWebs", on the line to select a web server or disk location type in the `servername:portnumber` of the web to edit then click OK.
  - — Select the web you wish to edit from the list of webs on the host machine.
- You need to provision each additional web. You can do this from the client side, with the FrontPage client provided the client has the right authorization (the administrator username and password) for the root web. You can also provision user webs from the server side by using the program `fpsrvadm.exe` to set the password for an individual web. You need to make sure that the new FrontPage web does not inherit the administrator username and password from the root web.

- Locate the `fpadmin.htm` file, typically in the `\admin\cgi` directory (97 and 98) or `\admcgi` directory (2000) of your FrontPage program directory. You can use this to configure your FrontPage web.
- Users can edit the local web that is displayed when FrontPage is started, but they must have a valid user ID and password to modify it.

Installing FrontPage97 Server Extensions on Unix/Linux Systems

The installation process on a Unix/Linux system requires you to have the appropriate file permissions and directories set up beforehand. The extensions require a specific directory structure, which is discussed later in this section. After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

These installation instructions are for the standalone FrontPage Server Extensions that are in a tarred file that is downloadable from the Microsoft FrontPage web site or the Ready-To-Run Software website.

Important You need to be logged in as the root user to perform the install. The root user must have write permission for the `/usr/local` directory even if this is not the directory where you want to install the extensions. This is because if you choose not to install the extensions in the /`usr/local` directory, a soft link is added automatically between `/usr/local` and the directory you wish to use.

To install the extensions, perform the following steps:

1. Log in as the root user so you can install the FrontPage Server Extensions from the tar file:

   ```
   cd /usr/local
   ```

2. Untar the downloaded file.

   This creates a `/usr/local/FrontPage/version2.0` directory and installs several other new directories under the document root directory. For example, for the FrontPage 97 extensions on a Solaris platform, you untar the vt20.solaris.tar.z file:

   ```
   tar xvf /usr/tmp/vt20.solaris.tar
   ```

3. Change directories to `/usr/local/frontpage/version2.0`.

   ```
   cd frontpage/version2.0
   ```

4. Create a directory named `/extensions` and move the `_vti_bin` directory into it.

   ```
   mv _vti_bin extensions
   ```

5. Install the WPP kit to `/usr/local/frontpage/version2.0`.

   For Solaris, use this code:

   ```
   tar xvf /usr/tmp/wpp.solaris.tar
   ```

6. Rename the directory /executables (`/usr/local/frontpage/version2.0/executables`) to `/_vti_bin`:

   ```
   mv executables _vti_bin
   ```

7. Move the file `fpsrvadm.suid.exe` to the `/bin` directory:

   ```
   mv fpsrvadm.suid.exe bin
   ```

8. Run the `fp_install.sh` shell program and follow the on-screen instructions, which ask for the information described in the following table.

   When you are prompted for the name of the server configuration file, enter the pathname of your server's `magnus.conf` file.

Table 16.21 Installation parameter information

-fpdir <dir>	default	FrontPage Directory
-httpdconfdir <dir>	default	Directory where server's configuration file is located
-web <webname>	required	Web where the Server Extensions are being installed (/ for root web)
-user <webowner>	required	User ID of the web owner
-group <webgroup>	optional	GroupID of the web owner

Table 16.21 Installation parameter information

-host <host>		Name of virtual host where the Server Extensions are being installed. The host specified should be the same as that specified by the Virtual Host directive in the server's `httpd.conf` file.
-admuser <fpadmin>	required	FrontPage Administrator user name
-admpass <fppass>	required	FrontPage Administrator password
-admaddr <ipaddr>	optional	IP address restriction of FrontPage Administrator. If not IP address mask is specified, the FrontPage Administrator will have access from all IP addresses.

Installing the Server Extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page's contents are parsed to expand FrontPage components, such as Include components and Substitution components, create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation process also updates the text indices and recalculates the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

During installation, the install shell modifies or adds the following files and directories:

- Modifies `magnus.conf`
- Creates a configuration file named `/usr/local/frontpage/`*`hostname`*`:`*`port`*`.cnf`
- Modifies the server's configuration file (`obj.conf`) to add `ObjectType` directives, marking three of the `/_vti_` directories as containing executables.
- Adds seven subdirectories under the server's document root:

```
/_vti_bin (contains shtml.exe)
/_vti_bin/_vti_adm (contains admin.exe)
/_vti_bin/_vti_aut (contains author.exe)
/_vti_cnf
/_vti_pvt
/_private
/_vti_log
/_vti_txt
/images
```

- Creates `.nsconfig` files in the `/_vti_bin`, `/_vti_adm`, `/_vti_aut` and the document root directories.

Once you have completed the installation process, you must perform the following administrative tasks:

1. Execute the `fpsrvwin.exe` file to set the server port, test the extensions, install the extensions to other virtual servers, and update extensions.

2. A remote machine must have the FrontPage 97, 98, or 2000 program installed (Macintosh or Windows only).

   Once the FrontPage program is started, the user is prompted for the name of a server to edit or open.

3. If the user wants to edit a web on a different machine, click on "MoreWebs" on the line to select a web server or disk location type in the *`servername:portnumber`* of the web to edit. Choose OK.

4. Select the proper web from the list of webs on the host machine to edit.

Installing FrontPage98 Server Extensions on Unix/Linux Systems

These installation instructions are for the stand-alone FrontPage Server Extensions that are in a tarred file that is downloadable from the Microsoft FrontPage web site or the Ready-To-Run Software website.

After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

Important You need to be logged in as the root user to perform the install. The root user must have write permission for the `/usr/local` directory even if this is not the directory where you want to install the extensions. This is because if you choose not to install the extensions in the `/usr/local` directory, a soft link is added automatically between `/usr/local` and the directory you wish to use.

To install the extensions, perform the following steps:

1. Log in as the root user so you can install the FrontPage Server Extensions from the tar file.
2. Type `cd /usr/local`, or `cd` to the directory where the two downloaded files (`fp30.solaris.tar.Z` and `fp_install.sh`) are located.
3. Run the `fp_install.sh` shell program and follow the on-screen instructions, which ask for parameter information.

   When you are prompted for the name of the server configuration file, enter the pathname of your server's `magnus.conf` file.

Installing the Server Extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page's contents are parsed to expand FrontPage components, such as Include components and Substitution components, create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation process also updates the text indices and recalculate the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

During installation, the install shell modifies or adds the following files and directories:

- Modifies `magnus.conf`
- Creates a configuration file named `/usr/local/frontpage/`*`hostname`*`:`*`port`*`.cnf`
- Modifies the server's configuration file (`obj.conf`)to add `ObjectType` directives, marking three of the `/_vti_` directories as containing executables.

- Adds seven subdirectories under the server's document root:

```
/_vti_bin (contains shtml.exe)
/_vti_bin/_vti_adm (contains admin.exe)
/_vti_bin/_vti_aut (contains author.exe)
/_vti_cnf
/_vti_pvt
/_private
/_vti_log
/_vti_txt
/images
```

Creates .`nsconfig` files in the `/_vti_bin`, `/_vti_adm`, `/_vti_aut` and the document root directories.

Installing FrontPage2000 Server Extensions on Unix/Linux Systems

These installation instructions are for the stand-alone FrontPage Server Extensions that are in a tarred file that is downloadable from the Microsoft FrontPage web site or the Ready-To-Run Software website.

After installation, you must perform some additional administrative tasks for setting permissions and accessing specific webs.

Important You need to be logged in as the root user to perform the install. The root user must have write permission for the `/usr/local` directory even if this is not the directory where you want to install the extensions. This is because if you choose not to install the extensions in the `/usr/local` directory, a soft link is added automatically between `/usr/local` and the directory you wish to use.

To install the extensions, perform the following steps:

1. Log in as the root user so you can install the FrontPage Server Extensions from the tar file.
2. Type `cd /usr/local`, or `cd` to the directory where the two downloaded files (`fp40.solaris.tar.Z` and `fp_install.sh`) are located.
3. Run the `fp_install.sh` shell program and follow the on-screen instructions, which ask for parameter information.

When you are prompted for the name of the server configuration file, enter the pathname of your server's `magnus.conf` file.

Installing the Server Extensions on each FrontPage web may take several minutes and may increase the CPU load on your computer. If this is a new installation of the FrontPage Server Extensions, each page's contents are parsed to expand FrontPage components, such as Include components and Substitution components, create a hyperlink map of the FrontPage web, and extract page titles and base URLs.

The installation process also updates the text indices and recalculate the links in the Web, adds a FrontPage administration account, password, and IP address restriction, and reminds the web administrator to restart the server if new `ObjectType` directives were added to the `obj.conf` file.

During installation, the install shell modifies or adds the following files and directories:

- Modifies `magnus.conf`
- Creates a configuration file named `/usr/local/frontpage/`*`hostname`*`:`*`port`*`.cnf`
- Modifies the server's configuration file (`obj.conf`)to add `ObjectType` directives, marking three of the `/_vti_` directories as containing executables.
- Adds seven subdirectories under the server's document root:

```
/_vti_bin (contains shtml.exe)
/_vti_bin/_vti_adm (contains admin.exe)
/_vti_bin/_vti_aut (contains author.exe)
/_vti_cnf
/_vti_pvt
/_private
/_vti_log
/_vti_txt
/images
```

Creates `.nsconfig` files in the `/_vti_bin`, `/_vti_adm`, `/_vti_aut` and the document root directories.

Further Information

Additional detailed information can be obtained from Microsoft's FrontPage web site:

```
http://www.microsoft.com
```

For Unix/Linux only, information can also be obtained from the Ready-to-Run Software web site:

```
http://www.rtr.com
```

Glossary

Access Control Entries (ACEs) A hierarchy of rules which the web server uses to evaluate incoming access requests.

Access Control List (ACL) A collection of ACEs. An ACL is 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.

admpw The username and password file for the Enterprise Administrator Server superuser.

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 (CA) 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.

Certificate revocation list (CRL) CA list, provided by the CA, of all revoked certificates.

Compromised key list (CKL) A list of key information about users who have compromised keys. The CA also provides this list.

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.

daemon (Unix) A background process responsible for a particular system task.

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.

Enterprise Administration Server — A web-based server that contains the Java and JavaScript forms you use to configure all of your Netscape Enterprise Servers

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.

FORTEZZA — An encryption system used by U.S. government agencies to manage sensitive but unclassified information.

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 process or 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 daemon or 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.

inittab (Unix) A Unix file listing programs that need to be restarted if they stop for any reason It ensures that a program runs continuously. Because of its location, it is also called `/etc/inittab`. This file isn't available on all Unix systems.

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.

JavaServer Pages Extensions that enable all JavaServer page metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. JavaServer pages, are reusable Java applications that run on a web server rather than in a web browser.

Java Servlets Extensions that enable all Java servlet metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. Java servlets are reusable Java applications that run on a web server rather than in a web browser.

last-modified header The last modification time of the document file, returned in the HTTP response from the server.

`magnus.conf` The main Enterprise Server configuration file. This file contains global server configuration information (such as, port, security, and so on). This file sets the values for variables that configure the server during initialization. Enterprise Sever reads this file and executes the variable settings on startup. The server does not read this file again until it is restarted, so you must restart the server every time you make changes to this file.

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.

mime.types The MIME (Multi-purpose Internet Mail Extension) type configuration file. This file maps file extensions to MIME types, to enable the server to determine the type of content being requested. For example, requests for resources with

.`html` extensions indicate that the client is requesting an HTML file, while requests for resources with `.gif` extensions indicate that the client is requesting an image file in GIF format.

MTA Message Transfer Agent. You must define your server's MTA Host to use agent services on your server.

Netscape Console A Java application that provides server administrators with a graphical interface for managing all Netscape servers from one central location anywhere within your enterprise network. From any installed instance of Netscape Console, you can see and access all the Netscape servers on your enterprise's network to which you have been granted access rights.

NIS (Unix) Network Information Service. A system of programs and data files that Unix machines use to collect, collate, and share specific information about machines, users, file systems, and network parameters throughout a network of computers.

network management station (NMS) A machine users can use to remotely manage a network. A managed device is anything that runs SNMP such as hosts, routers, and Netscape/iPlanet servers. An NMS is usually a powerful workstation with one or more network management applications installed.

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.

`obj.conf` The server's object configuration file. This file contains additional initialization information, settings for server customization, and instructions that the server uses to process requests from clients (such as browsers). Enterprise Server reads this file every time it processes a client request.

password file (Unix) A file on Unix machines that stores Unix user login names, passwords, and user ID numbers. It is also known as `/etc/passwd`, because of where it is kept.

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.

public information directories (Unix) Directories not inside the document root that are in a Unix user's home directory, or directories that are under the user's control.

Quality Feedback Agent An error-handling mechanism that enables you to automatically send error information (stack and register dump) to Netscape.

RAM Random access memory. The physical semiconductor-based memory in a computer.

rc.2.d (Unix) A file on Unix machines that describes programs that are run when the machine starts. This file is also called `/etc/rc.2.d` because of its location.

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.

root (Unix) The most privileged user on Unix machines. The root user has complete access privileges to all files on the machine.

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.

superuser (Unix) The most privileged user available on Unix machines (also called root). The superuser has complete access privileges to all files on the machine.

Sym-links (Unix) Abbreviation for symbolic links, which is a type of redirection used by the Unix operating system. Sym-links let you create a pointer from one part of your file system to an existing file or directory on another part of the file system.

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 (Unix) A program on some Unix systems that shows the current state of system resource usage.

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).

uid (Unix) A unique number associated with each user on a Unix system.

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 `usr/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 (Windows NT) CGI programs written in a Windows-based programming language such as Visual Basic.

Index

Note that if you are looking for a reference to a specific Administration Server or Server Manager page, see the online help.

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X