Sun Java™ System

Web Proxy Server 4 .0.1 Administration Guide

2005Q4

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-3650-10

Contents

About This Guide

This guide describes how to configure and administer Sun Java™ System Web Proxy Server 4, formerly known as Sun™ ONE Web Proxy Server and iPlanet™ Web Proxy Server (and hereafter referred to as Sun Java System Web Proxy Server or just Proxy Server).

This preface contains the following sections:

- Who Should Use This Guide
- How This Guide is Organized
- Documentation Conventions
- Using the Documentation
- Contacting Sun Technical Support
- Feedback
- Third-Party Web Site References

Who Should Use This Guide

This guide is intended for information technology administrators in production environments. The guide assumes familiarity with the following areas:

- Performing basic system administration tasks
- Installing software
- Using web browsers
- Issuing commands in a terminal window

How This Guide is Organized

The guide is divided into parts, each of which addresses specific areas and tasks. The following table lists the parts of the guide and their contents.

Table 1 Guide Organization

Part	Description
Part 1 Server Basics	Provides an overview of the Proxy Server and its administration: • Chapter 1, "Introducing Sun Java System Web Proxy Server" • Chapter 2, "Administering Sun Java System Web Proxy Server"
Part 2 Using the Administration Server	Provides details about configuring Administration Server preferences, managing users and groups, securing the Proxy Server, and using clusters to share configurations among servers: • Chapter 3, "Setting Administration Preferences" • Chapter 4, "Managing Users and Groups" • Chapter 5, "Using Certificates and Keys" • Chapter 6, "Managing Server Clusters"
Part 3 Configuring and Monitoring the Proxy Server	Provides details about configuring server preferences, setting access control, and monitoring server activity: • Chapter 7, "Configuring Server Preferences" • Chapter 8, "Controlling Access to Your Server" • Chapter 9, "Using Log Files" • Chapter 10, "Monitoring Servers"

Table 1 Guide Organization

Part	Description
Part 4 Managing the Proxy Server	Provides details about concepts and tasks related to how the Proxy Server handles requests: • Chapter 11, “Proxying and Routing URLs” • Chapter 12, “Caching” • Chapter 13, “Filtering Content through the Proxy” • Chapter 14, “Using a Reverse Proxy” • Chapter 15, “Using SOCKS” • Chapter 16, “Managing Templates and Resources” • Chapter 17, “Using the Client Autoconfiguration File”
Part 5 Appendixes	Describes access control list (ACL) file syntax and tuning server performance: • Appendix A, “ACL File Syntax” • Appendix B, “Tuning Server Performance”

Documentation Conventions

The following table lists the documentation conventions used in this guide.

Table 2 Documentation Conventions

Element	Usage
File and directory paths	Given in UNIX® format, with forward slashes separating directory names.
Installation root directories	Indicated as *server_root*. The default installation directory is `/proxyserver4`.
Italic text	Titles, emphasis, terms, environment variables in paths, placeholders.
`monospace` text	Code examples, file names, path names, command names, programming language keywords, properties.

Using the Documentation

Sun Java System Web Proxy Server 4 documentation is available in HTML and PDF formats at:

http://docs.sun.com/app/docs/prod/s1.webproxys#hic

The following table lists the tasks and concepts described in each guide.

Table 3 Proxy Server Documentation

For Information About	See
The Proxy Server release: • Late-breaking information about the software and the documentation • New features • Supported platforms and environments • System requirements • Known issues and workarounds	*Release Notes*
Performing installation and migration tasks: • Installing Sun Java System Web Proxy Server • Migrating from version 3.6 to version 4	*Installation and Migration Guide*
Performing administration and management tasks: • Using the administration and command-line interfaces • Configuring server preferences • Managing users and groups • Monitoring and logging server activity • Using certificates and public key cryptography to secure the server • Controlling server access • Proxying and routing URLs • Caching • Filtering content • Using a reverse proxy • Using SOCKS	*Administration Guide* (and the online Help included with the product)
Creating custom Netscape Server Application Programmer’s Interface (NSAPI) plugins	*NSAPI Developer’s Guide*

Table 3 Proxy Server Documentation

For Information About	See
Editing configuration files	*Configuration File Reference*

Contacting Sun Technical Support

For technical questions about this product that are not answered in the product documentation, go to:

`http://www.sun.com/service/contacting`

Feedback

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click the link used to send comments. Be sure to provide the document title and part number in the online form.

Third-Party Web Site References

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Part I

Server Basics

Chapter 1

Introducing Sun Java System Web Proxy Server

This chapter provides a general overview of Sun Java™ System Web Proxy Server, including a brief description of what's new in this release and an overview of the web-based user interfaces used to administer, configure, and manage the Proxy Server.

This chapter contains the following sections:

- About Sun Java System Web Proxy Server
- New in This Release
- Getting Started

About Sun Java System Web Proxy Server

Sun Java System Web Proxy Server represents the HTTP caching and acceleration foundation for high performance Internet and Intranet environments. The Proxy Server is a powerful system for caching and filtering web content and boosting network performance, offering tight integration with the entire network infrastructure, cross-platform support, and centralized management capabilities. Acting as a network traffic manager, it distributes and manages information efficiently so network traffic and user wait times are reduced. The Proxy Server also helps ensure that users can access network resources safely and productively, providing a secure gateway for content distribution and serving as a control point for Internet traffic.

New in This Release

Sun Java System Web Proxy Server 4 includes the following enhancements:

- Modern HTTP core
- Support for Linux and the Solaris™ x86 platform
- Modern SSL (Secure Sockets Layer) support on all platforms
- Multi-threaded architecture on all platforms
- Improved administration, graphical user interface, and ease of management
- New NSAPI (Netscape Server Application Programmer's Interface) filters
- Increased LDAP (Lightweight Directory Access Protocol) performance
- Improved scalability and performance
- Improved content filtering
- Implementation of the `server.xml` configuration file

More information about new features and enhancements can be found in the Proxy Server *Release Notes*, available at:
http://docs.sun.com/app/docs/prod/s1.webproxys#hic

Getting Started

Sun Java System Web Proxy Server is administered and configured with the Administration Server and the Server Manager, web-based user interfaces that are accessed with a browser. The Administration Server is used to manage configuration that is common to all Proxy Server instances installed on your system, while the Server Manager is used to configure settings for individual server instances.

This section contains the following topics:

- Administration Server Overview
- Server Manager Overview
- Configuration Files
- Regular Expressions

NOTE Cookies must be enabled in your browser to run the CGI programs necessary for configuring your server.

Administration Server Overview

The Administration Server is a web-based user interface used to manage configuration that is common to all Sun Java System Web Proxy Server instances installed on your system.

After the Administration Server has been started (see "Starting the Administration Server""Starting the Administration Server" on page 33), the Administration Server is accessed by launching a browser and entering a URL. The URL is determined by the host name and port number specified during installation. For example: `http://myserver.mycorp.com:1234`.

More than one administrator can be granted access to the Administration Server. For more information about distributed administration, see "Allowing Multiple Administrators" on page 42.

To access the Administration Server

1. Launch a browser and enter the URL that reflects the host name and port number specified for the Administration Server during installation. For example: `http://myserver.mycorp.com:1234`.
2. When prompted, enter the user name and password specified during installation. The Administration Server's user interface displays.

Administration Server settings are organized by tabs that correspond to specific tasks. The following table lists the Administration Server tabs, and provides a brief description of what the tabs are used for.

Table 1-1 Administration Server Tabs

Tab	Used To
Servers	Manage, add, remove, migrate Proxy Servers
Preferences	Shutdown the Administration Server, edit listen sockets, configure superuser access, configure distributed administration (allow multiple administrators), customize and view access and error logs
Global Settings	Configure directory services, specify access control, configure SNMP master agent settings

Table 1-1 Administration Server Tabs

Tab	Used To
Users and Groups	Add and manage users, groups, and organizational units
Security	Create new trust databases, request and install VeriSign and other certificates, change the key-pair file password, view and manage installed certificates, add or replace Certificate Revocation Lists and Compromised Key Lists (CRLs and CKLs), manage CRLs and CKLs, migrate 3.x certificates
Cluster	Control remote servers in a cluster, add and remove remote servers, modify server information

Regardless of the tab or page you are on, the following buttons also display:

- **Version -** Displays version information about Sun Java System Web Proxy Server
- **Refresh -** Refreshes the current page
- **Help -** Displays online Help for the current page

For more information about using the Administration Server, see Chapter 2, "Administering Sun Java System Web Proxy Server" on page 33. Also see the online Help for Administration Server tabs and pages.

Server Manager Overview

The Server Manager is the web-based user interface used to start, stop, and configure individual instances of Sun Java System Web Proxy Server.

To access the Server Manager

1. Access the Administration Server as described in "Administration Server Overview" on page 27. The Administration Server displays on the Servers tab.
2. On the Manage Servers page, click the link for the server instance you want to manage. The Server Manager user interface displays.

Server Manager settings are organized by tabs that correspond to specific tasks. The following table lists the Server Manager tabs, and provides a brief description of what the tabs are used for.

Table 1-2 Server Manager Tabs

Tab	Used To
Preferences	Start and stop the server, view server settings, restore configuration information, configure system preferences, tune Proxy Server performance, add and edit listen sockets, manage MIME types, administer access control, configure ACL and DNS caches, configure DNS local subdomains, configure HTTP keep-alive settings, set cipher size
Routing	Enable and disable proxying, set routing preferences, forward client credentials, enable Java IP address checking, create and edit autoconfiguration files, set the connectivity mode, change the default FTP transfer mode, set the SOCKS name server IP address, configure HTTP request load balancing
SOCKS	Start and stop the SOCKS server, and create and manage SOCKS authentication, connection, and routing entries
URLs	View, create, and manage URL mappings and redirections
Caching	Set cache specifics, add and modify cache partitions, move sections among existing partitions, set the cache capacity, set the garbage collection mode, tune the cache, schedule garbage collection, tune garbage collection settings, configure caching for specific resources, enable the caching of local hosts, change the file cache settings, set cache batch updates, view information about recorded cached URLs, configure proxies in an ICP neighborhood, create and update the proxy array member list, configure proxy array members, view information in the PAT file
Filters	Create filter files, set content URL rewriting, set user-agent restriction and request blocking, suppress outgoing headers, set MIME filters and HTML tag filters, compress content on demand

Table 1-2 Server Manager Tabs

Tab	Used To
Server Status	View log files, archive logs, set log preferences, generate reports, monitor current activity, configure and control the SNMP subagent
Security	Create new trust databases, request and install VeriSign and other certificates, change the key-pair file password, view and manage installed certificates, add or replace Certificate Revocation Lists and Compromised Key Lists (CRLs and CKLs), manage CRLs and CKLs, migrate 3.x certificates
Templates	Create, remove, apply, and view templates, and remove resources

Regardless of the tab or page you are on, the following buttons also display:

- **Version -** Displays version information about Sun Java System Web Proxy Server
- **Refresh -** Refreshes the current page
- **Help -** Displays online Help for the current page

At times you may also see a Restart Required link below the Refresh button. This indicates that changes have been made for which a server restart is required. To apply the changes, click the link and specify the desired action.

For more information about using the Server Manager, see related tasks in this guide. Also see the online Help for Server Manager tabs and pages.

Configuration Files

The configuration and behavior of Sun Java System Web Proxy Server is determined by a set of configuration files. Settings configured in the administration interface are reflected in the configuration files. The files can also be edited manually.

The configuration files reside in the directory *instance_dir*/`config`, where *instance_dir* is the server instance. The `config` directory contains various configuration files that control different components. The exact number and names of the configuration files depend on which components have been enabled or loaded. This directory always contains four configuration files that are essential to server operation. The following table lists the four essential configuration files and their contents.

Table 1-3 Essential Configuration Files

File	Contains
`server.xml`	Most of the server configuration (new in this Proxy Server release)
`magnus.conf`	Global server initialization information
`obj.conf`	Instructions for handling requests from clients
`mime.types`	Information for determining the content type of requested resources

For detailed information about these and other configuration files, see the Proxy Server *Configuration File Reference*.

Regular Expressions

Regular expressions can be used to identify resources and configure the Proxy Server to handle requests from different URLs differently. Regular expressions can be specified as you perform a variety of tasks using the Administration Server and Server Manager user interfaces. For detailed information about the use of regular expressions, see Chapter 16, “Managing Templates and Resources” on page 349.

Chapter 2

Administering Sun Java System Web Proxy Server

This chapter introduces the basics of administering Sun Java System Web Proxy Server using the Administration Server. The Administration Server is a web-based user interface used to manage, add, remove, and migrate servers.

This chapter contains the following sections:

- Starting the Administration Server
- Stopping the Administration Server
- Running Multiple Proxy Servers
- Removing a Server Instance
- Migrating from Proxy Server 3.6

For details about configuring Administration Server preferences, see Chapter 3, “Setting Administration Preferences” on page 39. For details about administering multiple Proxy Servers using server clusters, see Chapter 6, “Managing Server Clusters” on page 121.

Starting the Administration Server

This section describes how to start the Administration Server on different platforms. For information about stopping the Administration Server, see “Stopping the Administration Server” on page 34.

To start the Administration Server on UNIX or Linux

- From the command line, go to *server_root*/`proxy-admserv` and type `./start` to start the Administration Server (or `./restart` to restart the Administration Server).

To start the Administration Server on Windows

- Use Start > Programs > Sun Microsystems > Sun Java System Web Proxy Server *version* > Start Admin

  -or-

  Use Control Panel > Administrative Tools > Services > Sun Java System Web Proxy Server 4.0 > Start

  -or-

- From a command prompt, go to *server_root*\`proxy-admserv` and type `startsvr.bat` to start the Administration Server (or `./restart` to restart the Administration Server).

After the Administration Server has been started, it can be accessed by launching a browser and entering a URL that reflects the host name and port number specified for the Administration Server during installation (for example, `http://myserver.mycorp.com:1234`). You will be prompted for a user name and password, both of which were also specified during installation.

More than one administrator can be granted access to the Administration Server. For more information about distributed administration, see “Allowing Multiple Administrators” on page 42.

Stopping the Administration Server

This section describes how to stop the Administration Server on different platforms. For information about starting the Administration Server, see “Starting the Administration Server” on page 33.

To stop the Administration Server on UNIX or Linux

- Access the Administration Server, click the Preferences tab, click the Shutdown Server link, and then click OK.

  -or-

- From the command line, go to *server_root*/`proxy-admserv/` and type `./stop`.

To stop the Administration Server on Windows

- Use the Sun Java System Proxy Server 4.0 Administration Server service in the Services window: Control Panel > Administrative Tools > Services

  -or-

- From the command prompt, go to *server_root*`\proxy-admserv` and type `stopsvr.bat`.

Running Multiple Proxy Servers

To run multiple Proxy Servers on your system multiple server instances must be installed and configured. The following procedure describes how to add server instances.

To install multiple server instances

1. Access the Administration Server.
2. On the Servers tab, click Add Server.
3. Provide the requested information and click OK. For more information about specific fields, see the online Help.
4. If desired, click the Configure Your New Server link on the Success page that displays after successfully adding a new server instance. The Server Manager interface displays, and is used to configure server instances.

Removing a Server Instance

The Administration Server can be used to remove Proxy Server instances. This process cannot be undone, so be sure you want to remove the server instance before performing the following procedure.

To remove a server instance

1. Access the Administration Server.
2. On the Servers tab, click Remove Server.
3. From the drop-down list, select the server instance you want to remove.
4. To proceed with removal, select the Confirming Server Removal checkbox and click OK.

Migrating from Proxy Server 3.6

Sun™ One Web Proxy Server 3.6 (also known as iPlanet™ Web Proxy Server) can be migrated to Sun Java System Web Proxy Server 4. The 3.6 server is preserved, and a new version 4 server with the same settings is created. For more information about migrating a server from version 3.6 to version 4, see the Proxy Server *Installation and Migration Guide*. Also see the online Help for migration-related pages in the Proxy Server user interface. For information about migrating certificates, see "Migrating Certificates" on page 86 in this guide.

Part 2

Using the Administration Server

Chapter 3

Setting Administration Preferences

This chapter describes how to configure administration preferences using the Administration Server. Cookies must be enabled in your browser to run the CGI programs necessary for configuring your server.

This chapter contains the following sections:

- Creating and Managing Listen Sockets
- Changing Superuser Settings
- Allowing Multiple Administrators
- Specifying Log File Options
- Using Directory Services
- Restricting Server Access
- SNMP Master Agent Settings

Creating and Managing Listen Sockets

Before the server can process a request, the request must be accepted by a listen socket and then directed to the correct server. When the Proxy Server is installed, one listen socket (`ls1`) is created automatically. This listen socket uses the IP address `0.0.0.0` and the port number specified as the Administration Server port number during installation.

Listen sockets are added, edited, and deleted using the Administration Server’s Edit Listen Sockets page. You must have at least one listen socket with which to access the server. You cannot delete a listen socket if it is the only one listed.

This section contains the following topics:

- Adding Listen Sockets
- Editing Listen Sockets
- Deleting Listen Sockets

Adding Listen Sockets

To add listen sockets

1. Access the Administration Server and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Click the New button.
4. Specify the settings and click OK. For more information about specific fields, see the online Help.

Editing Listen Sockets

To edit listen sockets

1. Access the Administration Server and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Click the link for the listen socket you want to edit, make the desired changes, and then click OK.

Deleting Listen Sockets

To delete listen sockets

1. Access the Administration Server and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Select the checkbox next to the listen socket you want to delete and click OK. You will be prompted to confirm deletion. You must have at least one listen socket with which to access the server. You cannot delete the listen socket if it is the only one listed.

Changing Superuser Settings

Superuser access can be configured for the Administration Server. These settings affect only the superuser account. If the Administration Server uses distributed administration, additional access controls must be configured for the permitted administrators.

CAUTION If Sun Java™ System Directory Server is used to manage users and groups, the superuser entry must be updated in the directory *before* changing the superuser user name or password. If you do not update the directory first, you will not be able to access the Users and Groups interface in the Administration Server. To fix this, you must either access the Administration Server with an administrator account that does have access to the directory, or update the directory using the Directory Server's console or configuration files.

To change superuser settings for the Administration Server

1. Access the Administration Server and click the Preferences tab.
2. Click the Control Superuser Access link.
3. Make the desired changes and click OK. For more information about specific fields, see the online Help.

The superuser's user name and password are kept in a file called `admpw`, located in *server_root*`/proxy-admserv/config`. The file has the format *username*`:`*password*. You can view this file to obtain the user name, but the password is encrypted and unreadable. If you forget the password, you can edit the `admpw` file and simply delete the encrypted password. You can then do the following:

1. Access the Administration Server with the user name and no password.
2. Click the Preferences tab.
3. Click the Control Superuser Access link.
4. Provide a new password and click OK.

CAUTION Because the `admpw` file can be edited, it is very important that the server computer be kept in a secure place and that access to its file system be restricted.

On UNIX and Linux systems, consider changing file ownership so that it is writable only by root or whatever system user runs the Administration Server daemon. On Windows systems, restrict file ownership to the user account used by the Administration Server.

Allowing Multiple Administrators

Multiple administrators can change specific parts of the server through distributed administration. A directory server must be installed before distributed administration can be enabled. The default directory service must be LDAP-based.

There are two levels of users for distributed administration: superuser and administrators.

- Superuser is the user listed in *server_root*`/proxy-admserv/config/admpw`. This is the user name and password specified during installation. This user has full access to all forms in the Administration Server except the Users and Groups forms, the access for which depends on the superuser having a valid account in an LDAP 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 configured for them, usually by the superuser. Administrators can perform limited administrative tasks and also make changes that affect other users, such as adding users or changing access control.

For more information about access control, see Chapter 8, “Controlling Access to Your Server” on page 149.

To enable distributed administration

1. Verify that a directory server is installed.
2. Access the Administration Server.
3. After a directory server has been installed, you may also need to create an administration group if you have not already done so. To create a group:
   a. Click the Users and Groups tab.
   b. Click the Create Group link.

c. Create an administrators group in the LDAP directory, and add the names of the users to whom you are granting permission to configure the Administration Server or any of the servers installed in its server root. For more information about specific fields, see the online Help.

   All users in the administrators group have full access to the Administration Server, but access control can be used to limit the servers and forms they are allowed to configure.

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

4. Click the Preferences tab.
5. Click the Configure Distributed Administration link.
6. Select Yes, specify the administrator group, and then click OK.

Specifying Log File Options

The Administration Server log files record data about the Administration Server, including the types of errors encountered and information about server access. Viewing these logs allows you to monitor server activity and troubleshoot problems. The type and format of the data recorded in the Administration Server logs is specified using the many options on the Log Preferences pages. You can choose the Common Logfile Format, which provides a fixed amount of information about the server, or you can create a custom log file format that better suits your requirements.

To access the Administration Server Log Preferences pages, click the Preferences tab, then click the Set Access Log Preferences or Set Error Log Preferences link. For detailed information about the log files and setting log file options, see Chapter 9, “Using Log Files” on page 181. Also see the online Help.

Viewing Log Files

Administration Server log files are located in *server_root*/`proxy-admserv/logs`. You can view both the error and access log through the Proxy Server administration console, or with a text editor.

The Access Log File

The access log file records information about requests to and responses from the server.

To view the access log file

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

For more information about specific fields, see the online Help. Also see Chapter 9, "Using Log Files" on page 181.

The Error Log File

The error log lists all 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 to log in but failed.

To view the error log file

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

For more information about specific fields, see the online Help. Also see Chapter 9, "Using Log Files" on page 181.

Using Directory Services

You can store and manage information such as user names and passwords in a single directory server using LDAP. You can also configure the server to allow users to retrieve directory information from multiple, easily accessible network locations. For more information about using directory services, see Chapter 4, "Managing Users and Groups" on page 47.

Restricting Server Access

When the Proxy Server evaluates an incoming request, access is determined based on a hierarchy of rules called access control entries (ACEs), and then matching entries are used to determine if the request should be allowed or denied. Each ACE specifies whether the server should continue to the next ACE in the hierarchy. The collection of ACEs is called an access control list (ACL).

Access control can be configured for access to the Administration Server and to specific resources within a server instance, such as files, directories, and file types. Access control to the Administration Server is configured from the Global Settings tab in the Administration Server. Access control for resources within a server instance is configured from the Preferences tab in the Server Manager. For more information about setting access control, see Chapter 8, "Controlling Access to Your Server" on page 149.

NOTE Distributed administration must be enabled before you can restrict server access. For more information, see "Allowing Multiple Administrators" on page 42.

SNMP Master Agent Settings

Simple Network Management Protocol (SNMP) is a protocol used to exchange data about network activity. This information is transferred between a network management station and the server through the use of subagents and master agents.

SNMP master agent settings are configured using the Global Settings tab in the Administration Server. The master agent is installed with the Administration Server. For detailed information about SNMP and agent settings, see Chapter 10, "Monitoring Servers" on page 207. Also see the online Help for master agent pages on the Global Settings tab in the Administration Server, and for the subagent pages on the Server Status tab in the Server Manager.

Chapter 4

Managing Users and Groups

This chapter describes how to add, delete, modify, and manage the users and groups that can access the Proxy Server.

This chapter contains the following sections:

- Accessing Information about Users and Groups
- About Directory Services
- Configuring Directory Services
- Creating Users
- Managing Users
- Creating Groups
- Managing Groups
- Creating Organizational Units
- Managing Organizational Units

Accessing Information about Users and Groups

The Administration Server provides access to application data about user accounts, group lists, access privileges, organizational units, and other user- and group-specific information.

User and group information is stored either in flat files in text format, or in a directory server such as Sun Java™ System Directory Server, which supports LDAP (Lightweight Directory Access Protocol). LDAP is an open directory access protocol that runs over TCP/IP (Transmission Control Protocol/Internet Protocol) and is scalable to a global size and millions of entries.

About Directory Services

A directory service enables all user information to be managed from a single source. With Proxy Server, three different types of directory services can be configured: LDAP, key file, and digest file.

If no other directory service has been configured, the first new directory service created is set to the value `default`, irrespective of its type. When a directory service is created, the *server_root*/`userdb/dbswitch.conf` file is updated with directory service details.

This section contains the following topics:

- LDAP Directory Services
- Key File Directory Services
- Digest File Directory Services

LDAP Directory Services

With an LDAP directory service, user and group information is stored in an LDAP-based directory server.

If the LDAP service is the default service, the `dbswitch.conf` file is updated as shown in the example below:

```
directory default
ldap://test22.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcom
default:binddn cn=Directory Manager
default:encoded bindpw YWRtaW5hZG1pbg==
```

If the LDAP service is a non-default service, the `dbswitch.conf` file is updated as shown in the example below:

```
directory ldap
ldap://test22.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcom
ldap:binddn cn=Directory Manager
ldap:encoded bindpw YWRtaW5hZG1pbg==
```

Key File Directory Services

A key file is a text file that contains the user's password in a hashed format and the list of groups to which the user belongs. The key file format can only be used when the intent is to use HTTP Basic authentication. For more information about this authentication method, see "Specifying Users and Groups" on page 166.

When a key file-based database is created, the `dbswitch.conf` file is updated as shown in the example below:

```
directory keyfile file
keyfile:syntax keyfile
keyfile:keyfile D:\test22\keyfile\keyfiledb
```

Digest File Directory Services

A digest file stores user and group information based on encrypted user name and password.

The digest file format is meant to support the use of HTTP Digest authentication but also supports Basic authentication, so it can be used for both authentication methods. For more information about these methods, see "Specifying Users and Groups" on page 166.

When a digest-based database is created, the `dbswitch.conf` file is updated as shown in the example below:

```
directory digest file
digest:syntax digest
digest:digestfile D:\test22\digest\digestdb
```

NOTE To configure distributed administration, the default directory service must be an LDAP-based directory service.

Configuring Directory Services

A directory service is created and configured on the Global Settings tab in the Administration Server. Users, groups, and organizational units are then created and managed on the Users and Groups tab in the Administration Server.

This section contains the following topics:

- Creating Directory Services
- Editing Directory Services

Creating Directory Services

To create directory services

1. Access the Administration Server and click the Global Settings tab.
2. Click the Configure Directory Service link.
3. From the Create New Service of Type drop-down list, select the type of directory service you want to create and click New. The configuration page for that directory service displays.
4. Provide configuration information, and then click Save Changes. For more information about specific fields, see the online Help.

NOTE If no other directory service has been configured, the first new directory service created is set to the value `default`, irrespective of its type.

Editing Directory Services

To edit directory services

1. Access the Administration Server and click the Global Settings tab.
2. Click the Configure Directory Service link.
3. Click the link for the directory service you want to edit, make the desired changes, and then click Save Changes. For more information about specific fields, see the online Help.

Understanding Distinguished Names (DNs)

The Users and Groups tab in the Administration Server is used to create or modify users, groups, and organizational units. A user is an individual in the LDAP database, such as an employee of your company. A group is two or more users who share a common attribute. An organizational unit is a subdivision within your organization that uses the `organizationalUnit` object class. Users, groups, and organizational units are described in greater detail later in this chapter.

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 user or group directory entries are changed. For example, DN information must be provided each time you create or modify directory entries, configure access controls, and configure user accounts for applications such as mail or publishing. The Users and Groups interface of the Proxy Server is used to create or modify DNs.

The following example represents a typical DN for an employee of Sun Microsystems:

```
uid=doe,e=doe@sun.com,cn=John Doe,o=Sun Microsystems Inc.,c=US
```

The abbreviations in this example mean the following:

- `uid` is the user ID
- `e` is the e-mail address
- `cn` is the user's common name
- `o` is the organization
- `c` is the country

DNs may include a variety of name-value pairs, and are used to identify both certificate subjects and entries in directories that support LDAP.

Using LDIF

If you do not currently have a directory, or you want to add a new subtree to an existing directory, you can use the directory server’s LDIF (Lightweight Directory Interchange Format) 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. You can add or edit entries using the `ldapmodify` command line utility (if available), along with the appropriate LDIF update statements.

To add entries to the database using LDIF, first define the entries in an LDIF file, then import the LDIF file from the directory server.

Creating Users

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

NOTE Be sure to protect server security by ensuring that users do not have unauthorized access to resources. Proxy Server uses an ACL-based authorization and authentication model. For more information about ACL-based security, see Chapter 8, "Controlling Access to Your Server" on page 149. For additional security information, also see Chapter 5, "Using Certificates and Keys" on page 77.

This section contains the following topics:

- Creating Users in LDAP-based Authentication Databases
- Creating Users in Key File Authentication Databases
- Creating Users in Digest File Authentication Databases

Creating Users in LDAP-based Authentication Databases

When user entries are added to an LDAP-based directory service, the services of an underlying LDAP-based directory server are used to authenticate and authorize users. This section lists guidelines to consider when using an LDAP-based authentication database, and describes how to add users through the Proxy Server Administration Server.

Guidelines for Creating LDAP-based User Entries

Consider the following guidelines when using the Proxy Server administration console to create new user entries in an LDAP-based directory service:

- If you enter a given name (or first name) and a surname, the user’s full name and user ID are automatically completed. 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, 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, unique user IDs are not ensured. If duplicate user IDs exist in your directory, the affected users will not be able to authenticate to the directory.
- The base DN specifies the distinguished name where directory lookups occur by default, and where all Proxy Server Administration Server entries are placed in your directory tree. A DN is the string representation for the name of an entry in a directory server.
- 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 are defined for your directory, you can specify where you want the new user to be placed using the Add New User To list on the Create User page in the Administration Server. The default location is your directory’s base DN, or root point.

Creating LDAP-based User Entries

To create a user entry, read the guidelines outlined in “Guidelines for Creating LDAP-based User Entries” on page 53, then perform the following procedure.

To create users in LDAP-based authentication databases

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Create User link.
3. Select the LDAP directory service from the drop-down list and click Select.

4. Enter the information on the page that displays. For more information about specific fields, see the online Help. Also see "Directory Server User Entries" on page 54.

5. Click Create to create the user entry, or Create and Edit to create the user entry and proceed to the edit page for the entry just created.

Directory Server User Entries

Notes about directory server user entries:

- 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 DN is:

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

  However, note that this format can be changed to a user ID (uid)-based distinguished name.

- The values on the user form fields are stored as LDAP attributes.

  The following table lists the fields and corresponding LDAP attributes that display when creating a new user in the Proxy Server interface.

Table 4-1 LDAP Attributes - Creating User Entries

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

The following table lists the fields and corresponding LDAP attributes that also display when editing the user entry.

Table 4-2 LDAP Attributes - Editing User Entries

User Field	LDAP Attribute
Title	`title`
Phone Number	`telephoneNumber`

Creating Users in Key File Authentication Databases

A key file is a text file that contains the user's password in a hashed format, and the list of groups to which the user belongs.

To create users in key file authentication databases

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Create User link.
3. Select the key file-based directory service from the drop-down list and click Select.
4. Enter the information on the page that displays, and then click Create User. For more information about specific fields, see the online Help.

Creating Users in Digest File Authentication Databases

A digest file authentication database stores user and group information in an encrypted form.

To create users in digest file authentication databases

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Create User link.
3. Select the digest file-based directory service from the drop-down list and click Select.
4. Enter the information on the page that displays, and then click Create User. For more information about specific fields, see the online Help.

NOTE The same realm string must be specified when creating an ACL that uses Digest authentication using the Proxy Server ACL user interface. For more information, see "Setting Access Control" on page 161.

Managing Users

User attributes are edited from the Manage Users page on the Administration Server Users and Groups tab. From this page you can find, change, rename, and delete user entries.

This section contains the following topics:

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

Finding User Information

Before you can edit a user entry you must first find and display the entry, as described in the following procedure.

To find user information

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Users link.
3. Select a directory service from the drop-down list and click Select. For key file or digest file directory services, a list of users displays. For LDAP-based directory services, search fields display.

4. Find user information:

For key file or digest file directory services, click the link for the user to display the edit page and make changes. For more information about specific fields, see the online Help.

For LDAP-based directory services, do the following:

a. In the Find User field, enter a descriptive value for the entry you want to edit. You can enter any of the following:

- A name. Enter a full or 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 are found. If no such entries are found, any entries that sound like the search string are found.
- A user ID. If you enter only a partial user ID, any entries that contain the string are returned.
- A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number are returned.
- An e-mail address. Any search string containing an at symbol (@) is assumed to be an e-mail address. If an exact match cannot be found, a search is performed to find all e-mail addresses that begin with the search string.
- An asterisk (*) to see all entries currently in your directory. You can achieve the same result by 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 drop-down menus in the Find All Users Whose section to narrow the results of your search. For more information, see "Building Custom Search Queries" on page 58.

b. 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 (topmost entry).

c. In the Format field, specify whether the output should be formatted for display on screen or for printing to a printer.

d. At any stage in this process, click the Find button. All users matching your search criteria will be displayed.

e. Click the link for the entry you want to display.

Building Custom Search Queries

For LDAP services, the Find All Users Whose section enables you to build a custom search filter. Use the fields to narrow search results returned by a Find User search.

The left drop-down list specifies the attribute on which the search will be based. The following tables lists the available search attribute options.

Table 4-3 Search Attribute Options

Option	Searches for a Match
Full name	Each entry's full name
Last name	Each entry's last name, or surname
User ID	Each entry's user ID
Phone number	Each entry's phone number
E-mail address	Each entry's e-mail address

The center drop-down list specifies the type of search to perform. The following tables lists the available search type options.

Table 4-4 Search Type Options

Option	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 a 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 (specifies an equality search). Use this option when you know the exact value of a user's attribute. For example, you know the exact spelling of the user's name.
Isn't	Returns all entries whose attribute value does not exactly match the search string. Use this option to find all users in the directory whose name is not "John Smith." Note that use of this option can cause an extremely large number of entries to be returned.
Sounds like	Causes an approximate, or phonetic, search to be performed. Use this option if you know an attribute's value but do not know the spelling. For example, you do not know if a user's name is spelled "Sarret," "Sarette," or "Sarett."
Starts with	Causes a substring search to be performed. Returns all entries whose attribute value starts with the specified search string. For example, you know a user's name starts with "Miles," but do not know the rest of the name.

Table 4-4 Search Type Options

Option	Description
Ends with	Causes a substring search to be performed. Returns all entries whose attribute value ends with the specified search string. For example, you know a user's name ends with "Dimaggio," but do not know the rest of the name.

The right text field is used to enter a search string. To display all user entries contained in the directory specified in the Look Within field, enter an asterisk (*) or leave this field blank.

Editing User Information

To edit user entries

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Users link.
3. Display the user entry as described in "Finding User Information" on page 56.
4. Make the desired changes. For more information about specific fields, see the online Help.

NOTE You may want to change an attribute value that is not displayed by the edit user page. In this situation, use the directory server `ldapmodify` command line utility, if available.

For information about changing a user's user ID, see "Renaming Users" on page 60.

Managing a User's Password

The following procedure describes how to change or create user passwords.

To change or create user passwords

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Users link.
3. Display the user entry as described in "Finding User Information" on page 56.

4. Make the desired changes. For more information about specific fields, see the online Help.

For LDAP databases you can also disable the user's password by clicking the Disable Password button on the page used to edit user password information (accessed from the Manage Users page). Doing this prevents the user from logging into a server, without your having to delete the user's directory entry. You can allow access for the user again by entering a new password.

Renaming Users

For LDAP databases, the rename feature changes only the user ID. All other fields are left intact. You cannot use the rename feature to move the entry from one organizational unit to another.

To rename user entries

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Users link.
3. Display the user entry as described in "Finding User Information" on page 56.
4. Click the Rename User button on the edit user page, enter the user ID on the page that displays, and click Save Changes.

NOTE You can specify that the Administration Server no longer retains the old values when an entry is renamed by setting the `keepOldValueWhenRenaming` parameter to `false` (the default). This parameter is found in the following file:

server_root/`proxy-admserv/config/dsgw-orgperson.conf`

Removing Users

To remove user entries

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Users link.
3. Display the user entry as described in "Finding User Information" on page 56.

4. Click Delete User (LDAP) or Remove User (key file and digest file).

Creating Groups

A group is an object that describes a set of objects in an LDAP database. A Sun Java System server group consists of users who share a common attribute. For instance, the set of objects might be a number of employees who work in the Marketing division of your company. These employees might belong to a group called Marketing.

For LDAP services, there are two ways to define the membership of a group: statically and dynamically. Static groups enumerate their member objects explicitly. A static group is a common name (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 enable you to use an 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 those employees are already in the LDAP database under `ou=Sales,o=Airius.com`, you would define a dynamic group with the following member URL:

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

This group would subsequently contain all objects that have a `uid` attribute in the tree below the `ou=Sales,o=sun` point. Thus, all Sales members.

For static and dynamic groups, members can share a common attribute from a certificate if the `memberCertDescription` is used. Note that this only applies if the ACL uses the SSL method.

Once a new group has been created, you can add users (members) to it.

This section contains the following topics:

- About Static Groups
- About Dynamic Groups

About Static Groups

For LDAP services, 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 does not change unless a user is added to or deleted from the group.

Guidelines for Creating Static Groups

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

- Static groups can contain other static or dynamic groups.
- If organizational units are defined for your directory, specify where you want the new group to be placed using the Add New Group To list on the Create Group page in the Administration Server interface. The default location is your directory's root point (topmost entry).
- For more information about editing groups, see "Editing Group Entries" on page 68.

Creating Static Groups

To create static groups

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Create Group link.
3. Select New Group from the Type of Group drop-down list, and then click Go.
4. Enter the information on the Create Group page. For more information about specific fields, see the online Help.
5. Click Create to create the group, or Create and Edit to create the group and proceed to the edit page for the group just created.

About Dynamic Groups

For LDAP services, Proxy 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 that 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.

How Dynamic Groups are Implemented

Proxy Server implements dynamic groups in the LDAP server schema as objectclass=groupOfURLs. A groupOfURLs class can have zero or more memberURL attributes, each of which is an LDAP URL that describes 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 a filter, but not a host name and port. This means you can only refer to objects on the same LDAP server. All scopes are supported. For more information about LDAP URLs, see "Guidelines for Creating Dynamic Groups" on page 64.

The DNs are included automatically, without having to add each individual to the group. The group changes dynamically, because Proxy 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 Proxy Server uses the cn attribute as the group name for ACLs.

The mapping from an ACL to an LDAP database is defined both in the dbswitch.conf 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 with 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).

NOTE Groups can be both 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

Using dynamic groups impacts server performance. If you are testing group membership, and the DN is not a member of a static group, Proxy Server checks all dynamic groups in the database's base DN. Proxy Server determines if each `memberURL` matches by checking its base DN and scope against the DN of the user, and then performs a base search using the user DN as base DN and the filter of the `memberURL`. This procedure can involve a large number of individual searches.

Guidelines for Creating Dynamic Groups

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

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

  `ldap:///`*base_dn*`?`*attributes*`?`*scope*`?(`*filter*`)`

  The following table lists the required parameters for the LDAP URL.

Table 4-5 Required Parameters for the LDAP URL

Parameter Name	Description
`base_dn`	The 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 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. This parameter is ignored for dynamic group membership checks.

Table 4-5 Required Parameters for the LDAP URL

Parameter Name	Description
`scope`	This parameter is required. 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.
`(filter)`	This parameter is required. Search filter to apply to entries within the specified scope of the search. If you are using the Administration Server interface, you must specify this attribute. The parentheses are required.

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

Continuing with the guidelines for creating dynamic groups:

- If organizational units are defined for your directory, specify where you want the new group to be placed using the Add New Group To list on the Create Group page in the Administration Server interface. The default location is your directory's root point (topmost entry).
- For more information about editing groups, see “Editing Group Entries” on page 68.

Creating Dynamic Groups

To create dynamic groups

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Create Group link.
3. Select Dynamic Group from the Type of Group drop-down list and click Go.
4. Enter the information on the Create Group page. For more information about specific fields, see the online Help.

5. Click Create to create the group, or Create and Edit to create the group and proceed to the edit page for the group just created.

Managing Groups

For LDAP services, the Administration Server enables you to edit groups and manage group memberships from the Manage Groups page on the Administration Server Users and Groups tab.

This section contains the following topics:

Finding Group Entries

Before you can edit a group entry you must first find and display the entry, as described in the following procedure.

To find group entries

1. Access the Administration Server and click the Users and Groups tab.

2. Click the Manage Groups link.

3. Enter the name of the group you want to find in the Find Group field. You can enter any of the following:

    ❍ A name. Enter a full or 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 are found. If no such entries are found, any entries that sound like the search string are found.

- ❍ An asterisk (*) to see all groups currently residing in your directory. You can achieve the same result by 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 Find All Groups Whose section to build a custom search filter and narrow the results of your search. For more information, see "The Find All Groups Whose Section" on page 67.

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 (topmost entry).
5. In the Format field, specify whether the output should be formatted for display on screen or for printing to a printer.
6. At any stage in this process, click the Find button. All groups matching your search criteria will be displayed.
7. Click the link for the entry you want to display.

The Find All Groups Whose Section

For LDAP services, the Find All Groups Whose section enables you to build a custom search filter. Use the fields in this section to narrow the search results that are otherwise returned by Find Group.

The left drop-down list specifies the attribute on which the search is based. The following options are available:

- **Name.** Searches each entry's full name for a match.
- **Description.** Searches each group entry's description for a match.

The center drop-down list specifies the type of search to perform. The following options are available:

- **Contains.** Causes a sub-string search to be performed. Entries with attribute values containing the specified search string are returned. For example, if you know a group's name probably contains the word "Administrator," use this option with the search string "Administrator" to find the group entry.
- **Is.** Causes an exact match to be found. Use this option when you know the exact value of a group's attribute. For example, you know the exact spelling of the group's name.

- **Isn't.** Returns all entries whose attribute value does not exactly match the search string. Use this option if you want to find all groups in the directory whose name does not contain "administrator." Be aware, however, that using this option can cause an extremely large number of entries to be returned.
- **Sounds like.** Causes an approximate, or phonetic, search to be performed. Use this option if you know an attribute's value, but are unsure of the spelling. For example, you do not know if a group's name is spelled "Sarret's list," "Sarette's list," or "Sarett's list."
- **Starts with.** Causes a sub-string search to be performed. Returns all entries whose attribute value starts with the specified search string. For example, you know a group's name starts with "Product," but do not know the rest of the name.
- **Ends with.** Causes a sub-string search to be performed. Returns all entries whose attribute value ends with the specified search string. For example, you know a group's name ends with "development," but do not know the rest of the name.

In the right text field, enter a search string. To display all group entries contained in the Look Within directory, enter an asterisk (*) or leave this field blank.

Editing Group Entries

To edit group entries

This procedure applies to LDAP services only.

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Groups link.
3. Locate the group you want to edit as described in "Finding Group Entries" on page 66.
4. Make the desired changes. For more information about specific fields and buttons, see the online Help.

NOTE You may want to change an attribute value that is not displayed by the group edit page. In this situation, use the directory server `ldapmodify` command line utility, if available.

Adding Group Members

To add members to a group

This procedure applies to LDAP services only.

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Groups link.
3. Locate and display the group you want to manage as described in "Finding Group Entries" on page 66, and click the Edit button next to Group Members. Any existing group members are listed on the page that displays. Search fields also display.
   - To add user entries to the list of members, Users must be selected in the Find drop-down list.
   - To add group entries to the group, Groups must be selected.
4. In the Matching text field, enter a search string. Enter any of the following options:
   - A name. Enter a full or partial name. All entries whose name matches the search string are 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 sound like the search string are found.
   - A user ID. If you enter only a partial user ID, any entries that contain the string are returned.
   - A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number are returned.
   - An e-mail address. Any search string containing an at symbol (@) is assumed to be an e-mail address. If an exact match cannot be found, a search is performed to find all e-mail addresses that begin with the search string.
   - Enter an asterisk (*) or leave this field blank to see all 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 Add to find all matching entries in the LDAP database and add them to the group. If the search returns any entries you do not want added to the group, click the corresponding checkbox in the Remove From List column. (Note that you can also construct a search filter to match the entries you want removed from the group, and then click Remove. For more information, see "Removing Entries from the Group Members List" on page 70.)

6. When the list of group members is complete, click Save Changes. The entries are added to the group member list.

Adding Groups to the Group Members List

For LDAP services, 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 69.

Removing Entries from the Group Members List

This procedure applies to LDAP services only.

To remove entries from the group members list

1. Access the Administration Server and click the Users and Groups tab.

2. Click the Manage Groups link.

3. Locate the group you want to manage as described in "Finding Group Entries" on page 66, and click the Edit button next to Group Members.

4. For each member that you want to remove from the list, click the corresponding checkbox in the Remove From List column. You can also construct a search filter to match the entries you want removed from the group, and then click Remove. For more information about creating a search filter, see "Adding Group Members" on page 69.

5. Click Save Changes. The entries are deleted from the group members list.

Managing Owners

For LDAP services, a group owners list is managed in the same way as a group members list.

The following table lists the topics in this guide that provide more information.

Table 4-6 Managing Owners

To	See
Add owners to the group	"Adding Group Members" on page 69
Add groups to the owners list	"Adding Groups to the Group Members List" on page 70
Remove entries from the owners list	"Removing Entries from the Group Members List" on page 70

Managing See Alsos

See Alsos are references to other directory entries that may be relevant to the current group. They enable users to easily find entries for people and other groups that are related to the current group. You manage See Alsos the same way you manage the group members list.

The following table lists the topics in this guide that provide more information.

Table 4-7 Managing See Alsos

To	See
Add users to See Alsos	"Adding Group Members" on page 69
Add groups to See Alsos	"Adding Groups to the Group Members List" on page 70
Remove entries from See Alsos	"Removing Entries from the Group Members List" on page 70

Renaming Groups

This procedure applies to LDAP services only. When you rename a group entry, only the group's name is changed. 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.

To rename groups

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Groups link and locate the group you want to manage as described in "Finding Group Entries" on page 66.
3. Click the Rename Group button, specify a new group name on the page that displays, and click Save Changes.

Removing Groups

This procedure applies to LDAP services only.

To remove groups

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Groups link.
3. Locate the group you want to manage as described in "Finding Group Entries" on page 66 and click Delete Group.

NOTE	Individual members of the group are not removed. Only the group entry is removed.

Creating Organizational Units

For LDAP services, an organizational unit can include a number of groups and usually represents a division, department, or other discrete entity. A DN can exist in more than one organizational unit.

To create organizational units

1. Access the Administration Server and click the Users and Groups tab.

2. Click the Create Organizational Unit link.
3. Enter the information and click Create. For more information about specific fields, see the online Help.

Notes about organizational units:

- 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

For LDAP services, organizational units are edited and managed from the Manage Organizational Units page on the Administration Server Users and Groups tab.

This section contains the following topics:

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

Finding Organizational Units

This procedure applies to LDAP services only.

To find organizational units

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Organizational Units link.
3. Enter the name of the unit you want to find in the Find Organizational Unit field. You can enter any of the following:

- A name. Enter a full or 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 are found. If no such entries are found, any entries that sound like the search string are found.
- An asterisk (*) to see all groups currently residing in your directory. You can achieve the same result by 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 drop-down menus in the Find All Units Whose section to narrow the results of your search. For more information, see "The Find All Units Whose Section" on page 74.

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 (topmost entry).
5. In the Format field, specify whether the output should be formatted for display on screen or for printing to a printer.
6. At any stage in this process, click the Find button. All organizational units matching your search criteria will be displayed.
7. Click the link for the entry you want to display.

The Find All Units Whose Section

For LDAP services, the Find All Units Whose section enables you to build a custom search filter. Use the fields in this section to narrow the search results that are otherwise returned by Find Organizational Unit.

The left drop-down list specifies the attribute on which the search is based. The following options are available:

- **Unit name.** Searches each entry's full name for a match.
- **Description.** Searches each organizational unit entry's description for a match.

The center drop-down list specifies the type of search to perform. The following options are available:

- **Contains.** Causes a sub-string search to be performed. Entries with attribute values containing the specified search string are returned. For example, if you know an organizational unit's name probably contains the word "Administrator," use this option with the search string "Administrator" to find the organizational unit entry.

- **Is.** Causes an exact match to be found. Use this option when you know the exact value of an organizational unit’s attribute. For example, you know the exact spelling of the organizational unit’s name.
- **Isn’t.** Returns all entries whose attribute value does not exactly match the search string. That is, use this option if you want to find all organizational units in the directory whose name does not contain "administrator." Be aware, however, that use of this option can cause an extremely large number of entries to be returned.
- **Sounds like.** Causes an approximate, or phonetic, search to be performed. Use this option if you know an attribute’s value, but are unsure of the spelling. For example, you do not know if an organizational unit’s name is spelled "Sarret’s list," "Sarette’s list," or "Sarett’s list."
- **Starts with.** Causes a sub-string search to be performed. Returns all entries whose attribute value starts with the specified search string. For example, you know an organizational unit’s name starts with "Product," but do not know the rest of the name.
- **Ends with.** Causes a sub-string search to be performed. Returns all entries whose attribute value ends with the specified search string. For example, you know an organizational unit’s name ends with "development," but do not know the rest of the name.

In the right text field, enter a search string. To display all organizational unit entries contained in the Look Within directory, enter an asterisk (*) or leave this field blank.

Editing Organizational Unit Attributes

This procedure applies to LDAP services only.

To edit organizational unit entries

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Organizational Units link.
3. Locate the organizational unit you want to edit as described in “Finding Organizational Units” on page 73.
4. Make the desired changes. For more information about specific fields, see the online Help.

NOTE You may want to change an attribute value that is not displayed by the organizational unit edit page. In this situation, use the directory server `ldapmodify` command line utility, if available.

Renaming Organizational Units

This procedure applies to LDAP services only. When you rename an organizational unit entry, only the organizational unit's name is changed. You cannot use the rename feature to move the entry from one organizational unit to another.

To rename organizational unit entries

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Organizational Units link.
3. Locate the organizational unit you want to edit as described in "Finding Organizational Units" on page 73.
4. Click the Rename button, enter the new organizational unit name on the page that displays, and click Save Changes.

Removing Organizational Units

This procedure applies to LDAP services only.

To delete organizational unit entries

1. Access the Administration Server and click the Users and Groups tab.
2. Click the Manage Organizational Units link.
3. Locate the organizational unit you want to delete as described in "Finding Organizational Units" on page 73.
4. Click the Delete button and then click OK in the resulting confirmation box.

Chapter 5

Using Certificates and Keys

This chapter describes the use of certificates and keys authentication to secure Sun Java System Web Proxy Server. Proxy Server incorporates the security architecture of all Sun Java System servers, and is built on industry standards and public protocols for maximum interoperability and consistency.

This chapter assumes familiarity with the basic concepts of public-key cryptography, including encryption and decryption, public and private keys, digital certificates, and encryption protocols. For more information, see *Introduction to SSL*. This document can be found at http://docs.sun.com/source/816-6156-10/index.htm

This chapter contains the following sections:

- Certificate-based Authentication
- Creating a Trust Database
- Requesting and Installing a VeriSign Certificate
- Requesting and Installing Other Server Certificates
- Migrating Certificates
- Managing Certificates
- Installing and Managing CRLs and CKLs
- Setting Security Preferences
- Using External Encryption Modules
- Setting Client Security Requirements
- Setting Stronger Ciphers
- Other Security Considerations

Certificate-based Authentication

Authentication is the process of confirming identity. In the context of network interactions, authentication is the confident identification of one party by another. Certificates are one way of supporting authentication.

A certificate consists of digital data that specifies the name of an individual, company, or other entity, and certifies that the public key included in the certificate belongs to that entity.

Both clients and servers can have certificates. Server authentication refers to the confident identification of a server by a client (identification of the organization assumed to be responsible for the server at a particular network address). Client authentication refers to the confident identification of a client by a server (identification of the person assumed to be using the client software). Clients can have multiple certificates, much like a person might have several different pieces of identification.

A certificate is issued and digitally signed by a Certificate Authority, or CA. The CA can be a company that sells certificates, or 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 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 *Introduction to SSL*.

For more information regarding supported certificate extensions, see *All About Certificate Extensions*. This document can be found at http://www.mozilla.org/projects/security/pki/nss/tech-notes/tn3.html

NOTE	A server certificate must be installed before encryption can be activated.

Creating a Trust Database

Before requesting a server certificate you must create a trust database. In Proxy Server, the Administration Server and each server instance can have its own trust database. The trust database should only be created on your local computer.

When you create the trust database, you specify a password to be used for a key-pair file. You also need this password to start a server using encrypted communications. For a list of guidelines to consider when choosing a password, see “Choosing Strong Passwords” on page 116.

In the trust database you create and store the public and private keys, referred to as your key-pair file. The key-pair file is used for SSL encryption. You use the key-pair file when you request and install your server certificate. The certificate is stored in the trust database after installation.

The key-pair file is stored encrypted in the following directory:

server_root/`alias/proxy-`*serverid*`-key3.db`

The Administration Server can have only one trust database. Each server instance can have its own trust database.

To create a trust database

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Create Database link.
3. Enter a password for the trust database.
4. Enter the password again and click OK.

Using password.conf

By default, the Proxy Server prompts the administrator for the key database password before starting up. To restart an unattended Proxy Server, you must save the password in a `password.conf` file. Do this only if your system is adequately protected, so that this file and the key databases are not compromised.

Typically, you cannot start a UNIX SSL-enabled server with the `/etc/rc.local` or the `/etc/inittab` files because the server requires a password before starting. Although you can start an SSL-enabled server automatically if you keep the password in plaintext in a file, this is not recommended. The server’s `password.conf` file should be owned by root or the user who installed the server, with only the owner having read and write access to the file.

On UNIX, leaving the SSL-enabled server's password in the `password.conf` file 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 the `password.conf` file.

On Windows, if you have an NTFS file system, you should protect the directory that contains the `password.conf` file by restricting access, even if you do not use the file. The directory should have read and write permissions for the Administration Server user and the Proxy Server user. Protecting the directory prevents others from creating a false `password.conf` file. You cannot protect directories or files on FAT file systems by restricting access to them.

Starting an SSL-enabled Server Automatically

To start an SSL-enabled server automatically

1. Make sure SSL is enabled.

2. Create a new `password.conf` file in the `config` subdirectory of the Proxy Server instance.

   - If you are using the internal PKCS #11 software encryption module included with the Proxy Server, enter the following information:
     `internal:`*your_ password*

   - If you are using a different PKCS #11 module (for hardware encryption or hardware accelerators), specify the name of the PKCS #11 module, followed by the password. For example:
     `nFast:`*your_ password*

You will always be prompted to supply a password when starting the Proxy Server, even after the `password.conf` file has been created.

Requesting and Installing a VeriSign Certificate

VeriSign is Proxy Server's preferred Certificate Authority. The company's technology simplifies the certificate request process. VeriSign has the advantage of being able to return its certificate directly to your server.

After creating a certificate trust database for your server, you can request a certificate and submit it to a CA (Certificate Authority). If your company has its own internal CA, request your certificate from that department. 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.

The Administration Server can have only one server certificate. Each server instance can have its own server certificate.

This section contains the following topics:

- Requesting a VeriSign Certificate

- Installing a VeriSign Certificate

Requesting a VeriSign Certificate

To request a VeriSign certificate

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Request VeriSign Certificate link.
3. Review the steps listed on the page that displays and click OK. The VeriSign Enrollment Wizard displays and walks you through the process.

Installing a VeriSign Certificate

To install a VeriSign certificate

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Install VeriSign Certificate link.
3. Unless you plan to use an external encryption module, select Internal from the Cryptographic Module drop-down list.
4. Enter your key-pair file password or PIN.
5. Select the Transaction ID to retrieve from the drop-down list and click OK.

Requesting and Installing Other Server Certificates

In addition to VeriSign, you can also request and install certificates from other Certificate Authorities. Your company or organization may provide its own internal certificates. This section describes how to request and install other types of server certificates.

This section contains the following topics:

- Required CA Information
- Requesting Other Server Certificates

- Installing Other Server Certificates

Required CA Information

Before you start the request process, make sure you know what information your CA requires. The format of the requested information varies by CA, but you might typically be asked to provide the information listed below. Note that most of this information is usually not required for certificate renewals.

- **Requestor name.** The name under which the certificate will be issued.
- **Telephone number.** The telephone number of the requestor.
- **Common name.** The fully qualified host name used in DNS lookups (for example, `www.example.com`).
- **Email address.** The business e-mail address used for correspondence between you and the CA.
- **Organization.** The official, legal name of your company, educational institution, organization, and so on. Most CAs require that this information be verified with legal documents, such as a copy of a business license.
- **Organizational unit.** A description of an organizational unit within your company.
- **Locality.** A description of the city, principality, or country for the organization.
- **State or Province.** The state or province in which the business is located.
- **Country.** The two-character abbreviation of your country name in ISO format. For example, the country code for the United States is US.

All 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 with greater detail and veracity to organizations or individuals who provide more thorough identification. 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.example.com` computer, but that you are a company that has been in business for three years, and have no outstanding customer litigation.

Requesting Other Server Certificates

To request other server certificates

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Request Certificate link.
3. 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. Specify how you want to submit the request for the certificate:
    - To submit the request using e-mail, select CA Email Address and enter the appropriate e-mail address for such requests.
    - To submit the request using the CA’s web site, select CA URL and enter the appropriate URL for such requests.
5. From the Cryptographic Module drop-down list, select the cryptographic module to be used for the key-pair file when requesting the certificate.
6. Enter the password for your key-pair file. This is the password specified when you created the trust database, unless a cryptographic module other than Internal was selected. The server uses the password to obtain 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. Enter your identification information, such as name and phone number. The format of this information varies by CA. Note that most of this information is usually not required for certificate renewals.
8. Double-check your work to ensure accuracy, and then click OK. The more accurate the information, the faster your certificate is likely to be approved. If your request is going to a certificate server, you will be prompted to verify the form information before the request is submitted.

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 was not tampered with during routing from your server computer to the CA. In the rare event that the request is tampered with, the CA usually contacts you by phone.

If you chose to e-mail the request, the server composes an e-mail message containing the request and sends the message to the CA. Typically, the certificate is then e-mailed to you. If you specified a URL to a certificate server, your server uses the URL to submit the request to the certificate server. You might get an e-mail response or a response by some other means, depending on the CA.

The CA notifies you if it agrees to issue you a certificate. In most cases, the CA sends your certificate using e-mail. If your organization is using a certificate server, you may be able to search for the certificate using the certificate server's forms.

NOTE Not everyone who requests a certificate from a commercial CA is given one. Many CAs require you to prove your identity before issuing a certificate. Also, it often can take anywhere from one day to several weeks to get approval. You are responsible for promptly providing all necessary information to the CA.

Install the certificate once you receive it. In the meantime, you can still use your Proxy Server without SSL.

Installing Other Server Certificates

When you receive your certificate from the CA, it is encrypted with your public key so that only you can decrypt it. Only by entering the correct password for your trust database can you decrypt and install your certificate.

There are three types of certificates:

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

A certificate chain is a hierarchical series of certificates signed by successive Certificate Authorities. A CA certificate identifies a Certificate Authority 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 If your CA does not automatically send you its certificate, request it. Many CAs include their certificate in the e-mail with your certificate, and both certificates are installed by your server at the same time.

When you receive a certificate from the CA, it is encrypted with your public key so that only you can decrypt it. The Proxy Server uses the key-pair file password you specify to decrypt the certificate when it is installed. You can either save the e-mail somewhere accessible to the server, or copy the text of the e-mail and be ready to paste the text into the Install Certificate form, as described in the following procedure.

To install other server certificates

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Install Certificate link.
3. Next to Certificate For, select the type of certificate to install:
   - This Server
   - Server Certificate Chain
   - Certification Authority

   For more information about specific settings, see the online Help.
4. Select the cryptographic module from the drop-down list.
5. Enter the key-pair file password.
6. Enter a certificate name only if you selected Server Certificate Chain or Certification Authority in Step 3.
7. Provide certificate information by doing one of the following:
   - Select Message Is In This File and then enter the full path name to the file that contains the CA certificate.
   - Select Message Text (with headers) and then copy and paste the contents of the CA certificate. Be sure to include the Begin Certificate and End Certificate headers, including the beginning and ending hyphens.
8. Click OK.
9. Select either:

- Add Certificate if you are installing a new certificate.
- Replace Certificate if you are installing a certificate renewal.

The certificate is stored in the server's certificate database. For example:

server_root/`alias/proxy-`*serverid*`-cert8.db`

Migrating Certificates

When migrating from Sun ONE Web Proxy Server 3.6 (also known as iPlanet Web Proxy Server) to Sun Java System Web Proxy Server 4, your files, including your trust and certificate databases, are updated automatically.

Make sure that the Proxy Server 4 Administration Server has read permissions on the old 3.x database files. The files are *alias*`-cert.db` and *alias*`-key.db`, located in the *3.x_server_root*/`alias` directory.

Key-pair files and certificates are migrated only if security is enabled for your server. You can also migrate keys and certificates by themselves using the Migrate 3.x Certificates option under the Security tab in the Administration Server and the Server Manager. For information about specific settings, see the online Help.

In previous versions, a certificate and key-pair file were referred to by an alias that could be used by multiple server instances. The Administration Server managed all of the aliases and their constituent certificates. In Sun Java System Web Proxy Server 4, the Administration Server and each server instance has its own certificate and key-pair file, referred to as a trust database instead of an alias.

The trust database and its constituent certificates are managed from the Administration Server for the Administration Server itself, and from the Server Manager for server instances. The certificate and key-pair database files are now named after the server instance that uses them. If, in the previous version, multiple server instances shared the same alias, when migrated the certificate and key-pair file are renamed for the new server instance.

The entire trust database associated with the server instance is migrated. All CAs listed in your previous database are migrated to the Proxy Server 4 database. If duplicate CAs occur, use the previous CA until it expires. Do not attempt to delete duplicate CAs.

Proxy Server 3.x certificates are migrated to the supported Network Security Services (NSS) format. The certificate is named according to the Proxy Server page from which it was accessed (that is, from the Administration Server Security tab or the Server Manager Security tab).

To migrate a certificate

1. From your local computer, access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Migrate 3.x Certificates link.
3. Specify the root directory where the 3.6 server is installed.
4. Specify the alias for this computer.
5. Enter the administrator's password and click OK.

Using the Built-in Root Certificate Module

The dynamically loadable root certificate module included with Proxy Server contains the root certificates for many CAs, including VeriSign. The root certificate module allows you to upgrade your root certificates to newer versions in a much easier way. In the past you were required to delete the old root certificates one at a time, and then install the new ones, one at a time. To install well-known CA certificates, you can now simply update the root certificate module file to a newer version as it becomes available through future versions of the Proxy Server.

Because the root certificate is implemented as a PKCS #11 cryptographic module, you can never delete the root certificates it contains, and the option to delete will not be offered when managing these certificates. To remove the root certificates from your server instances, you can disable the root certificate module by deleting the following in the server's `alias` file:

- `libnssckbi.so` (on most UNIX platforms)
- `nssckbi.dll` (on Windows)

If later you want to restore the root certificate module, you can copy the extension from *server_root*`/bin/proxy/lib` (UNIX) or *server_root*`\bin\proxy\bin` (Windows) back into the `alias` subdirectory.

You can modify the trust information of the root certificates. The trust information is written to the certificate database for the server instance being edited, not back to the root certificate module itself.

Managing Certificates

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

To manage certificates

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Manage Certificates link.
   - If you are managing a certificate for a default configuration using the internal cryptographic module, a list of all installed certificates with their type and expiration date is displayed. All certificates are stored in the directory *server_root*/`alias`.
   - If you are using an external cryptographic module, such as a hardware accelerator, you must first enter your password for each specific module and click OK. The certificate list will update to include certificates in the module.
3. Click the name of the certificate you want to manage. A page displays with management options for that type of certificate. Only CA certificates allow you to set or unset client trust. Some external cryptographic modules will not allow certificates to be deleted.
4. Specify the desired action. The following options are available:
   - Delete certificate or Quit for certificates obtained internally
   - Set client trust, Unset server trust, or Quit for CA certificates

Certificate information includes the owner and who issued it. Trust settings allow you to set client trust or unset server trust. For LDAP server certificates, the server must be trusted.

Installing and Managing CRLs and CKLs

Certificate revocation lists (CRLs) and compromised key lists (CKLs) make known any certificates and keys that either client or server users should no longer trust. If data in a certificate changes, such as when 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. Contact your specific CA to obtain these lists.

This section contains the following topics:

- Installing CRLs or CKLs
- Managing CRLs and CKLs

Installing CRLs or CKLs

To install CRLs or CKLs

1. Obtain a CRL or CKL from your CA and download it to a local directory.
2. Access either the Administration Server or the Server Manager and click the Security tab.
3. Click the Install CRL / CKL link.
4. Select either:
   - Certificate Revocation List
   - Compromised Key List
5. Enter the full path name to the associated file and click OK. The Add Certificate Revocation List or Add Compromised Key List page displays, listing CRL or CKL information. If a CRL or CKL already exists in the database, a Replace Certificate Revocation List or Replace Compromised Key List page displays.
6. Add or replace the CRL or CKL.

Managing CRLs and CKLs

To manage CRLs and CKLs

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Manage CRL / CKL link. The Manage Certificate Revocation Lists / Compromised Key Lists page displays, listing all installed CRLs and CKLs and their expiration dates.
3. Select a certificate from either the Server CRLs or Server CKLs list.
4. Select Delete CRL or Delete CKL to delete the CRL or CKL, or Quit to return to the management page.

Setting Security Preferences

When you have a certificate, you can begin securing your server. Sun Java System Web Proxy Server provides many security elements, which are discussed in this section.

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. Proxy Server supports the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) encryption protocols.

A cipher is a cryptographic algorithm (a mathematical function) used for encryption or decryption. SSL and TLS protocols contain numerous cipher suites. Some ciphers are stronger and more secure than others. Generally speaking, the more bits a cipher uses, the harder it is to decrypt the data.

In any two-way encryption process, both parties must use the same ciphers. Because a number of ciphers are available, you must enable your server for those most commonly used.

During a secure connection, the client and the server agree to use the strongest cipher they can both have for communication. You can choose ciphers from the SSL 2.0, SSL 3.0, and TLS protocols.

NOTE Improvements to security and performance were made after SSL 2.0. Do not use SSL 2.0 unless you have clients that are incapable of using SSL 3.0. Client certificates are not guaranteed to work with SSL 2.0 ciphers.

The encryption process alone is not enough to secure your server's confidential information. 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. Information encrypted with a public key can be decrypted only with the associated private key. The public key is published as part of a certificate. Only the associated private key is safeguarded.

For a description of the various cipher suites and more information about keys and certificates, see *Introduction to SSL*.

To specify which ciphers your server can use, select them in the list in the Proxy Server user interface. Unless you have a compelling reason not to use a specific cipher, you should select them all (although you may not wish to enable ciphers with less than optimal encryption).

CAUTION Do not select Enable No Encryption, Only MD5 Authentication. If no other ciphers are available on the client side, the server defaults to this setting and no encryption occurs.

This section contains the following topics:

- SSL and TLS Protocols
- Using SSL to Communicate with LDAP
- Tunneling SSL Through the Proxy Server
- Configuring SSL Tunneling
- Enabling Security for Listen Sockets
- Configuring Security Globally

SSL and TLS Protocols

Proxy Server supports the SSL and TLS protocols for encrypted communication. SSL and TLS are application independent, and higher-level protocols can be layered transparently on them.

SSL and TLS protocols support a variety of ciphers used to authenticate the server and client to each other, transmit certificates, and establish session keys. Clients and servers may support different cipher suites, or sets of ciphers, depending on factors such as which protocol they support, company policies on encryption strength, and government restrictions on export of encrypted software. Among other functions, the SSL and TLS handshake protocols determine how the server and client negotiate which cipher suites they will use to communicate.

Using SSL to Communicate with LDAP

You should require your Administration Server to communicate with LDAP using SSL.

To enable SSL on your Administration Server

1. Access the Administration Server and click the Global Settings tab.
2. Click the Configure Directory Service link.

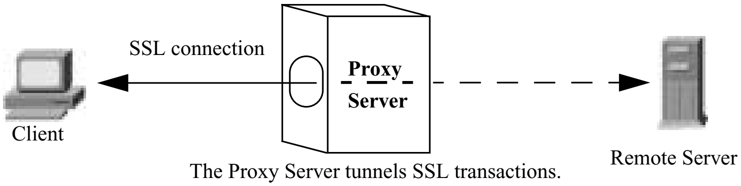
3. In the table that displays, click the link for the directory service. The Configure Directory Service page displays. If the LDAP-based directory service has not yet been created, select LDAP Server from the Create New Service of Type drop-down list, and then click New to configure the directory service. For more information about the specific fields that display for an LDAP-based directory service, see the online Help.

4. Select Yes to use SSL for connections, and then click Save Changes.

Tunneling SSL Through the Proxy Server

When you are running a Proxy Server (proxy) in the forward direction and a client requests an SSL connection to a secure server through the proxy, the proxy opens a connection to the secure server and then simply copies data in both directions without intervening in the secure transaction. This process is known as SSL tunneling, and is illustrated in the following figure.

Figure 5-1 With an SSL connection, the Proxy Server cannot view the data it transfers.



To use SSL tunneling with HTTPS URLs, the client must support both SSL and HTTPS. HTTPS is implemented using SSL with normal HTTP. Clients without HTTPS support can still access HTTPS documents using the Proxy Server's HTTPS proxying capability.

SSL tunneling is a lower-level activity that does not affect the application level (HTTPS). SSL tunneling is just as secure as SSL without proxying. The existence of the proxy in between does not in any way compromise security or reduce the functionality of SSL.

With SSL, the data stream is encrypted, so the proxy has no access to the actual transaction. Consequently, the access log cannot list the status code or the header length received from the remote server. This also prevents the proxy, or any other third party, from eavesdropping on the transactions.

Because the proxy never sees the data, it cannot verify that the protocol spoken between the client and the remote server is SSL. This means the proxy also cannot prevent other protocols from being passed through. You should restrict SSL connections to only well-known SSL ports, namely port 443 for HTTPS and 563 for SNEWS, as assigned by the Internet Assigned Numbers Authority (IANA). If there are sites that run the secure server on some other port, you can make explicit exceptions to allow connections to other ports on certain hosts. This is done using the `connect://.*` resource.

The SSL tunneling capability is actually a general, SOCKS-like capability that is protocol independent, so you can also use this feature for other services. Proxy Server can handle SSL tunneling for any application with SSL support, not just the HTTPS and SNEWS protocols.

Configuring SSL Tunneling

The following procedure describes how to configure your Proxy Server to tunnel SSL.

To configure SSL tunneling

1. Access the Server Manager for a server instance and click the Routing tab.

2. Click the Enable / Disable Proxying link.

3. Select the `connect://.*.443` resource from the drop-down list. The `connect://` method is an internal proxy notation and does not exist outside of the proxy. See the following description in "Technical Details for SSL Tunneling" on page 94 for more information about `connect`. To allow connections to other ports, you can use similar URL patterns in a template. For more information about templates, see Chapter 16, "Managing Templates and Resources" on page 349.

4. Select Enable Proxying Of This Resource and click OK.

CAUTION If the proxy is misconfigured, it is possible to abuse the SSL proxy to achieve telnet hopping. Someone can use the proxy to make it appear that a telnet connection is coming from the proxy host, rather than the actual connecting host. This is why you must allow no more ports than absolutely necessary, and use access control on your proxy (restricting the client hosts).

Technical Details for SSL Tunneling

Internally, SSL tunneling uses the `CONNECT` method with the destination host name and port number as a parameter followed by an empty line:

```
CONNECT energy.example.com:443 HTTP/1.0
```

A successful response from the Proxy Server would be the following, followed by an empty line:

```
HTTP/1.0 200 Connection established
Proxy-agent: Sun-Java-System-Web-Proxy-Server/4.0
```

The connection is then set up between the client and the remote server, and data can be transferred in both directions until either closes the connection.

Internally, to benefit from the typical configuration mechanism based on URL patterns, the host name and port number (`energy.example.com:443`) are automatically mapped into a URL such as this:

```
connect://energy.example.com:443
```

`connect://` is only an internal notation used by Proxy Server to make configuration easier and uniform with other URL patterns. Outside of the Proxy Server, `connect` URLs do not exist, and if the Proxy Server receives such a URL from the network, it marks it as invalid and refuses to service the request.

Enabling Security for Listen Sockets

You can secure your server’s listen sockets by doing the following:

- Turning the security on
- Selecting a server certificate for the listen socket
- Selecting ciphers

Turning Security On

You must turn security on before you can configure the other security settings for your listen socket. You can turn security on when you create a new listen socket or edit an existing one.

To turn security on when creating listen sockets

1. Access either the Administration Server or the Server Manager and click the Preferences tab.

2. Click the Add Listen Socket link.
3. Enter the required information. To turn security on, select Enabled from the Security drop-down list, and then click OK. Note that if a server certificate has not been installed, your only choice will be Disabled. For more information about specific settings, see the online Help.

NOTE Use the Edit Listen Sockets link to configure the security settings after a listen socket has been created.

To turn security on when editing listen sockets

1. Access either the Administration Server or the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Click the link for the listen socket you want to edit.
4. To turn security on, select Enabled from the Security drop-down list, and then click OK. Note that if a server certificate has not been installed, your only choice will be Disabled.

Selecting Server Certificates for Listen Sockets

You can configure listen sockets in either the Administration Server or the Server Manager to use server certificates you have requested and installed.

NOTE At least one certificate must be installed.

To select a server certificate for a listen socket

1. Access either the Administration Server or the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Click the link for the listen socket you want to edit.
4. To turn security on, select Enabled from the Security drop-down list, and then click OK. Note that if a server certificate has not been installed, your only choice will be Disabled.
5. After selecting Enabled and clicking OK, select a server certificate from the drop-down Server Certificate Name list for the listen socket, and then click OK.

Selecting Ciphers

To protect the security of your Proxy Servers, you should enable SSL. You can enable the SSL 2.0, SSL 3.0, and TLS encryption protocols and select the various cipher suites. The SSL and TLS protocols can be enabled on the listen socket for the Administration Server. Enabling SSL and TLS on a listen socket for the Server Manager sets those security preferences for specific server instances. At least one certificate must be installed.

NOTE Enabling SSL on a listen socket applies only in a reverse proxy scenario. That is, only when the Proxy Server is configured to perform reverse proxying.

The default settings allow the most commonly used ciphers. Unless you have a compelling reason for not using a specific cipher suite, you should select them all. For more information regarding specific ciphers, see *Introduction to SSL*.

The default and recommended setting for TLS Rollback is Enabled. This configures the server to detect man-in-the-middle version rollback attack attempts. Setting this to Disabled may be required for interoperability with some clients that incorrectly implement the TLS specification.

Note that disabling TLS Rollback leaves connections vulnerable to version rollback attacks. Version rollback attacks are a mechanism by which a third party can force a client and server to communicate using an older, less secure protocol such as SSL 2.0. Because there are known deficiencies in the SSL 2.0 protocol, failing to detect version rollback attack attempts makes it easier for a third party to intercept and decrypt encrypted connections.

To enable SSL and TLS

1. Access either the Administration Server or the Server Manager and click the Preferences tab.

2. Click the Edit Listen Sockets link, and then click the link for the listen socket you want to edit. For a secure listen socket, the available cipher settings are displayed.

NOTE If security is not enabled on the listen socket, no SSL and TLS information is listed. To work with ciphers, ensure that security is enabled on the selected listen socket. For more information, see "Enabling Security for Listen Sockets" on page 94.

3. Select the checkboxes corresponding to the required encryption settings and click OK.

NOTE	Select both TLS and SSL 3.0 for Netscape Navigator 6.0. For TLS Rollback also select TLS, and make sure both SSL 3.0 and SSL 2.0 are disabled.

Once SSL has been enabled on a server, its URLs use `https` instead of `http`. URLs that point to documents on an SSL-enabled server have this format:

`https://`*servername*`.`*domain*`.`*dom*`:`*port*

For example, `https://admin.example.com:443`.

If you use the default secure HTTP port (443), you do not need to enter the port number in the URL.

Configuring Security Globally

Installing an SSL-enabled server creates directive entries in the `magnus.conf` file (the server's main configuration file) for global security parameters.

To set values for SSL configuration file directives

1. Access the Server Manager for a server instance.
2. Ensure that security is enabled for the listen socket you want to configure. For more information, see "Enabling Security for Listen Sockets" on page 94.
3. Manually edit the `magnus.conf` file and enter values for the following settings:
   - `SSLSessionTimeout`
   - `SSLCacheEntries`
   - `SSL3SessionTimeout`

These SSL configuration file directives are described below. For more information about `magnus.conf`, see the Proxy Server *Configuration File Reference*.

SSLSessionTimeout

The `SSLSessionTimeout` directive controls SSL 2.0 session caching.

Syntax

`SSLSessionTimeout` *seconds*

where *seconds* is the number of seconds until a cached SSL 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.

SSLCacheEntries

Specifies the number of SSL sessions that can be cached.

SSL3SessionTimeout

The `SSL3SessionTimeout` directive controls SSL 3.0 and TLS session caching.

Syntax

`SSL3SessionTimeout` *seconds*

where *seconds* is the number of seconds until a cached SSL 3.0 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.

Using External Encryption Modules

Proxy Server supports the following methods of using external cryptographic modules such as smart cards or token rings:

- PKCS #11
- FIPS-140

You must add the PKCS #11 module before activating the FIPS-140 encryption standard.

This section contains the following topics:

- Installing the PKCS #11 Module
- FIPS-140 Standard

Installing the PKCS #11 Module

Proxy Server supports Public Key Cryptography Standard (PKCS) #11, which defines the interface used for communication between SSL and PKCS #11 modules. PKCS #11 modules are used for standards-based connectivity to SSL hardware accelerators. Imported certificates and keys for external hardware accelerators are stored in the `secmod.db` file, which is generated when the PKCS #11 module is installed. The file is located in the *server_root*/`alias` directory.

Using modutil to Install PKCS #11 Modules

You can install PKCS #11 modules in the form of `.jar` files or object files using the `modutil` tool.

To install PKCS #11 modules using modutil

1. Make sure all servers, including the Administration Server, have been stopped.
2. Go to the *server_root*/`alias` directory containing the databases.
3. Add *server_root*/`bin/proxy/admin/bin` to your PATH.
4. Locate `modutil` in *server_root*/`bin/proxy/admin/bin`.
5. Set the environment. For example:
   - On UNIX: `setenv`

     `LD_LIBRARY_PATH` *server_root*`/bin/proxy/lib:${LD_LIBRARY_PATH}`
   - On Windows, add it to the `PATH`

     `LD_LIBRARY_PATH` *server_root*`/bin/proxy/bin`

     You can find the PATH for your computer listed under: *server_root*/`proxy-admserv/start`.
6. Enter the command: `modutil`. The options will be listed.
7. Perform the actions required.

   For example, to add the PKCS #11 module in UNIX, enter:

   `modutil -add` (*name of PKCS#11 file*) `-libfile` (*your libfile for PKCS #11*) `-nocertdb -dbdir .` (*your db directory*)

Using pk12util

Using `pk12util` allows you to export certificates and keys from your internal database and import them into an internal or external PKCS #11 module. You can always export certificates and keys to your internal database, but most external tokens will not allow you to export certificates and keys. By default, `pk12util` uses certificate and key databases named `cert8.db` and `key3.db`.

Exporting with pk12util

To export a certificate and key from an internal database

1. Go to the *server_root*/`alias` directory containing the databases.
2. Add *server_root*/`bin/proxy/admin/bin` to your PATH.
3. Locate `pk12util` in *server_root*/`bin/proxy/admin/bin`.
4. Set the environment. For example:
    - On UNIX: `setenv`

      `LD_LIBRARY_PATH/`*server_root*`/bin/proxy/lib:${LD_LIBRARY_PATH}`
    - On Windows, add it to the `PATH`

      `LD_LIBRARY_PATH` *server_root*`/bin/proxy/bin`

      You can find the PATH for your computer listed under: *server_root*/`proxy-admserv/start`.
5. Enter the command: `pk12util`. The options will be listed.
6. Perform the actions required.

   For example, in UNIX enter:

   ```
   pk12util -o certpk12 -n Server-Cert [-d /server/alias] [-P https-test-host]
   ```
7. Enter the database password.
8. Enter the `pkcs12` password.

Importing with pk12util

To import a certificate and key into an internal or external PKCS #11 module

1. Go to the *server_root*/`alias` directory containing the databases.
2. Add *server_root*/`bin/proxy/admin/bin` to your PATH.
3. Locate `pk12util` in *server_root*/`bin/proxy/admin/bin`.

4. Set the environment. For example:

   ❍ On UNIX: `setenv`

     `LD_LIBRARY_PATH/`*server_root*`/bin/proxy/lib:${LD_LIBRARY_PATH}`

   ❍ On Windows, add it to the `PATH`

     `LD_LIBRARY_PATH` *server_root*`/bin/proxy/bin`

     You can find the PATH for your computer listed under: *server_root*`/proxy-admserv/start`.

5. Enter the command: `pk12util`. The options will be listed.

6. Perform the actions required.

   For example, in UNIX enter:

   ```
   pk12util -i pk12_sunspot [-d certdir][-h “nCipher”][-P
   https-jones.redplanet.com-jones-]
   ```

   `-P` must follow the `-h` and be the last argument.

   Enter the exact token name including capital letters and spaces between quotation marks.

7. Enter the database password.

8. Enter the `pkcs12` password.

Starting the Server with an External Certificate

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 edit the `server.xml` file or specify the certificate name as described below.

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 server certificate installed on an external smartcard reader called `smartcard0` would be named `smartcard0:Server-Cert`.

To start a server with a certificate installed in an external module, you must specify the certificate name for the listen socket on which it runs.

Selecting the Certificate Name for a Listen Socket

To select the certificate name for a listen socket

If security is not enabled on the listen socket, certificate information will not be listed. To select a certificate name for a listen socket, you must first ensure that security is enabled on the listen socket. For more information, see "Enabling Security for Listen Sockets" on page 94.

1. Access either the Administration Server or the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Click the link for the listen socket you want to associate with a certificate.
4. Select a server certificate from the Server Certificate Name drop-down list for the listen socket and click OK. The list contains all internal and external certificates installed.

You could also tell the server to start with that server certificate instead, by manually editing the `server.xml` file. Change the `servercertnickname` attribute in the `SSLPARAMS` to:

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

To find 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.

NOTE If you did not create a trust database, one will be created for you when you request or install a certificate for an external PKCS #11 module. The default database created has no password and cannot be accessed. Your external module will work, but you will not be able to request and install server certificates. If a default database has been created without a password, use the Create Database page on the Security tab to set the password.

FIPS-140 Standard

The PKCS #11 APIs enable communication with software or hardware modules that perform cryptographic operations. Once PKCS #11 is installed on your Proxy Server, you can configure the server to be FIPS-140 compliant (FIPS stands for Federal Information Processing Standards). These libraries are included only in SSL 3.0.

To enable FIPS-140

1. Install the plug-in following the FIPS-140 instructions.
2. Access either the Administration Server or the Server Manager and click the Preferences tab.
3. Click the Edit Listen Sockets link. For a secure listen socket, the Edit Listen Sockets page displays the available security settings.

NOTE To work with FIPS-140, ensure that security is enabled on the selected listen socket. For more information, see "Enabling Security for Listen Sockets" on page 94.

4. Select Enabled from the SSL Version 3 drop-down list, if not already selected.
5. Select the appropriate FIPS-140 cipher suite and click OK:
   - Enable Triple DES with 168 bit encryption and SHA authentication (FIPS)
   - Enable DES with 56 bit encryption and SHA authentication (FIPS)

Setting Client Security Requirements

After you have performed all of the steps to secure your servers, additional security requirements can be set for your clients.

Client authentication is not essential to an SSL connection, but it does give extra assurance that encrypted information is being sent to the correct parties. You can use client authentication in a reverse proxy to make sure that your content server does not share information with unauthorized proxies or clients.

This section contains the following topics:

- Requiring Client Authentication
- Client Authentication in a Reverse Proxy
- Setting Up Client Authentication in a Reverse Proxy
- Mapping Client Certificates to LDAP
- Using the certmap.conf File

Requiring Client Authentication

You can enable the listen sockets for your Administration Server and each server instance to require client authentication. When client authentication is enabled, the client's certificate is required before the server sends a response to a query.

Proxy Server supports authenticating client certificates by matching the CA in the client certificate with a CA trusted for signing client certificates. You can view a list of CAs trusted for signing client certificates in the Manage Certificates page on the Security tabs.

You can configure the Proxy Server to refuse any client that does not have a client certificate from a trusted CA. To accept or reject trusted CAs, client trust must be set for the CA. For more information, see "Managing Certificates" on page 87.

Proxy Server logs an error, rejects the certificate, and returns a message to the client if the certificate has expired. You can also view which certificates have expired from the Manage Certificates page.

You can configure your server to gather information from the client certificate and match 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. To learn how to do this, "Mapping Client Certificates to LDAP" on page 107.

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 (ACLs). For more information, see "Using Access Control Files" on page 158.

To require client authentication

1. Access either the Administration Server or the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Click the link for the listen socket for which you are requiring client authentication.
4. Use the Client Authentication drop-down list to require client authentication for the listen socket, and then click OK.

Client Authentication in a Reverse Proxy

In a reverse proxy, you can configure client authentication according to any one of the following scenarios:

- **Proxy-Authenticates-Client.** This scenario enables you to allow access to all clients with acceptable certificates, or to allow access to only those clients that have acceptable certificates and are recognized users on the access control list for your Proxy Server.
- **Content-Server-Authenticates-Proxy.** This scenario enables you to make sure that your content server is actually connecting with your Proxy Server and not some other server.
- **Proxy-Authenticates-Client and Content-Server-Authenticates-Proxy.** This scenario provides the maximum security and authentication for your reverse proxy.

For information about how to configure these scenarios, see "Setting Up Client Authentication in a Reverse Proxy" on page 105.

Setting Up Client Authentication in a Reverse Proxy

Client authentication in a secure reverse proxy provides further insurance that your connections are secure. The following instructions explain how to configure client authentication according to the scenario you choose.

NOTE Each scenario assumes that you have both a secure Client-to-Proxy connection and a secure Proxy-to-Content-Server connection.

Proxy-Authenticates-Client

To configure the Proxy-Authenticates-Client scenario

1. Follow the directions for configuring the secure Client-to-Proxy and secure Proxy-to-Content Server scenario in "Setting up a Reverse Proxy" in Chapter 14, "Using a Reverse Proxy" on page 317

2. Access the Server Manager for a server instance and click the Preferences tab.

3. Click the Edit Listen Sockets link, and then click the link for the desired listen socket in the table that displays. (Use the Add Listen Socket link to configure and add listen sockets.)

4. Specify client authentication requirements:

   To permit access to all users with valid certificates:

   - In the Security section, use the Client Authentication setting to require client authentication on this listen socket. Note that if a server certificate has not been installed, this setting will not be visible.

   To permit access to only those users who have both valid certificates and are specified as acceptable users in access control:

   a. In the Security section, leave the Client Authentication setting set to off. Note that if a server certificate has not been installed, this setting will not be visible.

   b. On the Server Manager Preferences tab for this server instance, click the Administer Access Control link.

   c. Select an ACL, and then click the Edit button. The Access Control Rules For page displays (authenticate first, if prompted).

   d. Turn access control on (select the Access control Is On checkbox if not already selected).

   e. Set your Proxy Server to authenticate as a reverse proxy. For more information, see “Setting up a Reverse Proxy” on page 321.

   f. Click the Rights link for the desired access control rule, specify access rights in the lower frame, and then click Update to update this entry.

   g. Click the Users/Groups link. In the lower frame, specify users and groups, select SSL as the authentication method, and then click Update to update this entry.

   h. Click Submit in the upper frame to save your entries.

   For more information about setting access control, see Chapter 8, “Controlling Access to Your Server” on page 149.

Content Server-Authenticates-Proxy

To configure the Content Server-Authenticates-Proxy scenario

1. Follow the directions for configuring the secure Client-to-Proxy and secure Proxy-to-Content-Server scenario in “Setting up a Reverse Proxy” on page 321.

2. On your content server, turn client authentication on.

NOTE You can modify this scenario so that you have an unsecure client connection to the Proxy Server, a secure connection to the content server, and the content server authenticates the Proxy Server. To do so, you must turn encryption off and tell the proxy to initialize certificates only as described in the following procedure.

Proxy-Authenticates-Client and Content Server-Authenticates-Proxy

To configure the Proxy-Authenticates-Client and Content Server-Authenticates-Proxy scenario

1. Follow the directions for configuring the Proxy-Authenticates-Client scenario in "Proxy-Authenticates-Client" on page 105.

2. On your content server, turn client authentication on.

Mapping Client Certificates to LDAP

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

When the server receives a request from a client, it asks for the client's certificate before proceeding. Some clients send the client certificate to the server along with the request.

NOTE Before mapping client certificates to LDAP, you must also configure the required ACLs. For more information, see Chapter 8, "Controlling Access to Your Server" on page 149.

The server tries to match the CA to the list of trusted CAs in the Administration Server. If there is no match, Proxy Server ends the connection. If there is a match, the server continues processing the request.

After verifying the certificate is from a trusted CA, the server maps the certificate to an LDAP entry by doing the following:

- Mapping the issuer and subject DN from the client certificate to a branch point in the LDAP directory.

- Searching the LDAP directory for an entry that matches the information about the subject (end user) of the client certificate.

- (Optional) Verifying 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 the LDAP search is performed. The mapping file tells the server what values to take from the client certificate (such as the end user's name, e-mail address, and so on). The server uses these values to search for a user entry in the LDAP directory, but first the server must determine where in the LDAP directory to start the search. The certificate mapping file also tells the server where to start.

Once the server knows where to start the search and what to search for (first point, above), it performs the search in the LDAP directory (second point). 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.

The following table lists the expected search result behavior. Note that you can specify the expected behavior in the ACL. For example, you can specify that the Proxy Server accepts only you if the certificate match fails. For more information about how to set the ACL preferences, see "Using Access Control Files" on page 158.

Table 5-1 LDAP Search Results

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

Using the certmap.conf File

Certificate mapping determines how a server looks up a user entry in the LDAP directory. You can use `certmap.conf` to configure how a certificate, designated by name, is mapped to an LDAP entry. 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. Users can be authenticated based on user ID, e-mail, or any other value used in the `subjectDN`. Specifically, the mapping file defines the following information:

- Where in the LDAP tree the server should begin the search
- What certificate attributes the server should use as search criteria when searching for the entry in the LDAP directory
- Whether the server goes through an additional verification process

The certificate mapping file is found 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 and can be defined to whatever you prefer. However, *issuerDN* must exactly match the issuer DN of the CA that issued the client certificate. For example, the following two issuer DN lines differ only in the spaces separating the attributes, but the server treats these two entries as different:

```
certmap sun1 ou=Sun Certificate Authority,o=Sun,c=US
certmap sun2 ou=Sun Certificate Authority, o=Sun, c=US
```

TIP If you are using Sun Java System Directory Server and experiencing problems in matching the issuer DN, check the Directory Server error logs for useful information.

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 DNComps is set 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 is no 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 e-mail and user ID attributes (FilterComps=e,uid), the server searches the directory for an entry whose values for e-mail and user ID match the end user's information gathered from the client certificate. E-mail addresses and user IDs 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.

  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 e-mail address, whereas LDAP calls that attribute mail.

  The following table lists the attributes for x509v3 certificates.

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
email	E-mail address

- verifycert tells the server whether the client's certificate should be compared with the certificate found in the LDAP directory. It takes two values: on and off. Use this property only if your LDAP directory contains certificates. This feature is useful to ensure that 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 is not a standard LDAP attribute, so to use this property, you must extend the LDAP schema. For more information, see *Introduction to SSL*.

  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 no entries are found, 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 path name to a shared library or DLL. Use this property only if you create your own properties using the certificate API.
- InitFn is a property whose value is the name of an init function from a custom library. Use this property only if you create your own properties using the certificate API.

For more information about these properties, refer to the examples described in "Sample Mappings" on page 112.

Creating Custom Properties

The client certificate API can be used to create your own properties. 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=Sun Microsystems, c=US
default1:library /usr/sun/userdb/plugin.so
default1:InitFn plugin_init_fn
default1:DNComps  ou o c
default1:FilterComps l
default1:verifycert on
```

Sample Mappings

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

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 italicized text is replaced with the values from the subject's DN in the client certificate.

The server then uses the values for e-mail address and user ID from the certificate to search for a match in the LDAP directory. When an entry is found, the server verifies the certificate by comparing the one sent by the client to the one stored in the directory.

Example #2

The following example file has two mappings: one for default 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 receives 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 e-mail and user ID. 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 e-mail addresses. Also note that if the certificate is from the US Postal Service, the server verifies the certificate. Other certificates are not verified.

CAUTION 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 will not match because there is not 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 uses DNComps and FilterComps to search for matching entries. In this example, the server searches 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.

Setting Stronger Ciphers

The Set Cipher Size option on the Server Manager Preferences tab presents a choice of 168-, 128-, or 56-bit secret key size for access or no restriction. You can specify a file to be served when the restriction is not met. If no file is specified, Proxy Server returns a Forbidden status.

If you select a key size for access that is not consistent with the current cipher settings under Security Preferences, Proxy Server displays a popup dialog warning that you need to enable ciphers with larger secret key sizes.

The implementation of the key size restriction is based on an NSAPI `PathCheck` directive in `obj.conf`, 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 to be served if the restriction is not met.

`PathCheck` returns `REQ_NOACTION` if SSL is not enabled, or if the `secret-keysize` parameter is not specified. If the secret key size 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 key size restriction is not met, the SSL session cache entry for the current session is invalidated, so that a full SSL handshake occurs the next time the same client connects to the server.

NOTE The Set Cipher Size form removes any `Service fn=key-toosmall` directives found in an object when it adds a `PathCheck fn=ssl-check`.

To set stronger ciphers

1. Access the Server Manager for a server instance and click the Preferences tab.
2. Click the Set Cipher Size link.
3. From the drop-down list, select the resource to which to apply stronger ciphers, and then click Select. You can also specify a regular expression. For more information, see Chapter 16, “Managing Templates and Resources” on page 349.
4. Select the secret key size restriction:
   - 168 bit or larger
   - 128 bit or larger
   - 56 bit or larger
   - No restrictions
5. Specify the file location of the message to reject access, and click OK.

For more information about ciphers, see *Introduction to SSL*.

Other Security Considerations

Other security risks exist beyond 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. In addition to enabling encryption on your server, you should take extra security precautions. For example, put the server computer in a secure room, and do not allow individuals you do not trust to upload programs to your server. This section describes some of the key things you can do to make your server more secure.

This section contains the following topics:

- Limiting Physical Access
- Limiting Administration Access
- Choosing Strong Passwords
- Changing Passwords or PINs
- Limiting Other Applications on the Server
- Preventing Clients from Caching SSL Files
- Limiting Ports
- Knowing Your Server's Limits

Limiting Physical Access

This simple security measure is often forgotten. Keep the server computer in a locked room that only authorized people can enter. This prevents anyone from hacking the server computer itself. Also, protect your computer's administrative (root) password, if you have one.

Limiting Administration Access

If you use remote configuration, be sure to set 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 about clusters, see Chapter 6, "Managing Server Clusters" on page 121.

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

Choosing Strong 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. Your private key password is the next most important. 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 will remember but others will not guess. For example, you could remember *MCi12!mo* as "My Child is 12 months old!" An example of a bad password is your child's name or birthday.

Creating Hard-to-Crack Passwords

Some simple guidelines will help you create a stronger password. It is not necessary to incorporate all of the following rules in one password, but the more rules you use, the better your chances of making your password hard to crack. Some tips:

- Passwords should be 6-14 characters long
- Do not use illegal characters: *, ", or spaces
- Do not use dictionary words (any language)
- Do not make common letter substitutions, such as replacing E with 3, or L with 1
- Include characters from as many of these classes as possible:
  - Uppercase letters
  - Lowercase letters

- Numbers
- Symbols

Changing Passwords or PINs

It is a good practice to change your trust database/key-pair file password or PIN 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.

You should only change this password on your local computer. For a list of guidelines to consider when changing a password, see "Creating Hard-to-Crack Passwords" on page 116.

To change the trust database/key-pair file password

1. Access either the Administration Server or the Server Manager and click the Security tab.
2. Click the Change Key Pair File Password link.
3. From the Cryptographic Module drop-down list, select the security token on which you want to change the password. By default, this is Internal for the internal key database. If PKCS #11 modules are installed, all of the security tokens will be listed.
4. Enter your current password.
5. Enter your new password.
6. Enter it again and click OK.

Make sure your key-pair file is protected. The Administration Server stores key-pair files in the directory *server_root/*`alias`.

It is also important to know if the file is stored on backup tapes or otherwise available for someone to intercept. If so, you must protect your backups as completely as your server.

Limiting Other Applications on the Server

Carefully consider all applications that run on the same computer as the server. It is 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 can be programmed to run other possibly detrimental programs on the server computer.

UNIX and Linux

Carefully choose the processes started from `inittab` and `rc` scripts. Do not run `telnet` or `rlogin` from the server computer. You also should not have `rdist` on the server computer. This can distribute files but can also be used to update files on the server computer.

Windows

Carefully consider which drives and directories you share with other computers. Also, consider which users have accounts or Guest privileges. Be careful about what programs you put on your server, or allow others to install. Other people's programs might have security holes. Even worse, someone might upload a malicious program designed specifically to subvert your security. Always examine programs carefully before you allow them on your server.

Preventing Clients from Caching SSL Files

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

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

Limiting Ports

Disable any ports not used on the computer. 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 computer is to physically use the server's computer, which should already be in a restricted area.

Knowing Your Server’s Limits

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

Being aware of these limitations helps you understand 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 computer? What happens to those numbers after the SSL connection is terminated? Be sure to secure any information clients send to you through SSL.

Chapter 6

Managing Server Clusters

This chapter describes the concept of clustering Sun Java System Web Proxy Servers, and describes how clusters can be used to share configurations among servers.

This chapter contains the following sections:

About Server Clusters

A cluster is a group of Sun Java System Web Proxy Servers that can be administered from a single Administration Server. Each cluster must include one server designated as the master Administration Server.

By organizing servers into clusters you can:

- Create a central place for administering all Proxy 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 specific servers

Guidelines for Using Clusters

The following list provides guidelines for configuring groups of Proxy Servers into clusters:

- All servers to be included in a particular cluster must be installed prior to creating any clusters.
- All servers in a cluster must be the same type (UNIX or Windows). Clusters must be homogenous.
- All servers in a cluster must be Proxy Server version 4. Only Proxy Server version 4 servers are supported for addition to clusters.
- All Administration Servers must use the same protocol, HTTP or HTTPS. If you change the protocol of one Administration Server in a cluster, you must change the protocols for all Administration Servers. For more information, see "Modifying Server Information" on page 124.
- All cluster-specific Administration Servers must have the same user name and password as the master Administration Server. Distributed administration can be used to configure multiple administrators on each Administration Server.
- One cluster-specific Administration Server must be designated as the master Administration Server (it does not matter which server you choose).
- The master Administration Server must have access to each cluster-specific Administration Server. The master Administration Server retrieves information about all installed Sun Java System Web Proxy Servers.

Setting Up Clusters

Following are the general steps you will take to set up Proxy Server clusters.

To set up Proxy Server clusters

1. Install the Proxy Servers you want included in the cluster. Make sure the Administration Server for the cluster has a user name and password that the master Administration Server can use for authentication. You can do this by using the default user name and password, or by configuring distributed administration.

2. Install the Proxy Server that will contain the master Administration Server, making sure the user name and password matches the one set in Step 1.

3. Add a server to the cluster list. For more information, see "Adding Servers to a Cluster" on page 123.

4. Administer a remote server by accessing its Server Manager interface from the Control Cluster page, or by copying a configuration file from one server in the cluster to another.

Adding Servers to a Cluster

When a Proxy Server is added to a cluster, its Administration Server and port number are specified. If that Administration Server contains information about more than one server, all of its servers are added to the cluster. Individual servers can be removed at a later time.

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 remote servers to a cluster

1. Make sure that the master Administration Server is turned on.

2. Access the master Administration Server and click the Cluster tab.

3. Click the Add Server link.

4. Choose the protocol used by the remote Administration Server:

   - HTTP for a typical Administration Server

   - HTTPS for a secure Administration Server

5. Enter the fully qualified host name of the remote Administration Server as it appears in the `magnus.conf` file (for example, `plaza.example.com`).

6. Enter the port number for the remote Administration Server.

7. Enter the administrator user name and password for the remote Administration Server and click OK. The master Administration Server attempts to contact the remote server. If successful, you are prompted to confirm the addition of the server to the cluster.

NOTE When cluster control is enabled, the master of the cluster creates a number of files in the `proxy-`*serverid*`/config/cluster/`*server_name*`/proxy-`*serverid* directory for each slave in the cluster. These files are not configurable.

Modifying Server Information

Use the Modify Server option on the Cluster tab of the Administration Server only to update slave administration port information after it has been changed on the slave server. If you change the port number of a remote Administration Server in your cluster, you must also modify the information about that Administration Server stored in the cluster. Any other changes to the slave Administration Server require you to remove the server, and then add it back into the cluster after the changes have been made.

To modify information about servers in a cluster

1. Access the master Administration Server and click the Cluster tab.
2. Click the Modify Server link. Servers display, listed by unique server identifiers.
3. Select the server to be modified, make the desired changes, and then click OK.

Removing Servers from a Cluster

To remove servers from a cluster

1. Access the master Administration Server and click the Cluster tab.
2. Click the Remove Server link.
3. Select the remote server(s) to remove from the cluster and click OK. The removed server can no longer be accessed through the cluster. It can only be accessed through its own Administration Server.

Controlling Server Clusters

Proxy Server enables you to control the remote servers in your cluster by:

- Starting and stopping the servers
- Viewing their access and error logs
- Transferring configuration files (if the master Administration Server has more then one instance of the Proxy Server, files can be transferred from any of these servers to any slave added to the cluster). Note that clusters must be homogeneous. All servers in a cluster must be of the same type (that is, either UNIX or Windows). Transferring configuration files from a different platform may cause the server to hang or crash. The configuration files are:
    - `server.xml`
    - `magnus.conf`
    - `obj.conf`
    - `mime.types`
    - `socks5.conf`
    - `bu.conf`
    - `icp.conf`
    - `parray.pat`
    - `parent.pat`

To control servers in a cluster

1. Access the master Administration Server and click the Cluster tab.
2. Click the Control Cluster link.
3. Select the server(s) to control and make your desired selections. Click the Reset button at any time to reset elements to the values they contained before any changes were made.
    - Select Start, Stop, or Restart from the drop-down list and click Go. You will be prompted to confirm the action.
    - Select View Access or View Error from the drop-down list and enter the last number of lines you want to view in the log file. Click Go to display the information (click the View button in the Cluster Execution Report that displays).

- Transfer configuration files:
    - Select the configuration file you want to transfer
    - Select the server on which the file resides
    - Click Go to transfer the information

Part 3

Configuring and Monitoring the Proxy Server

Chapter 7

Configuring Server Preferences

This chapter describes the Proxy Server's system settings and tells you how to configure them. System settings affect the entire Proxy Server. The settings include options such as the user account the proxy server uses and the port to which it listens.

This chapter contains the following sections:

Starting the Proxy Server

This section describes how to start the Proxy Server on different platforms. Once the server is installed, it runs, listening for and accepting requests.

To start the Proxy Server from the administration interface

1. Access the Server Manager and click the Preferences tab.
2. Click the Start / Stop Server link. The Start / Stop Server page displays.
3. Click the On button.

The status of the server appears in the Start / Stop Server page.

To start the Proxy Server on UNIX or Linux

- From the command line, go to *server_root*`/proxy-server`*id* and type `./start` to start the Proxy Server.
- Use `start`. If you want to use this script with `init`, you must include the start command `prxy:2:respawn:`*server_root*`/proxy-server`*id*`/start -start -i in /etc/inittab`.

To start the Proxy Server on Windows

- Use Start > Programs > Sun Microsystems > Sun Java System Web Proxy Server *version* > Start Proxy Server
- Use Control Panel > Administrative Tools > Services > Sun Java System Web Proxy Server 4.0 (proxy-server*id*) > Start
- From a command prompt, go to *server_root*`\proxy-server`*id* and type `startsvr.bat` to start the Proxy Server.

Starting SSL-enabled Servers

To start an SSL-enabled server, a password is required. Although you can start an SSL-enabled server automatically if you keep the password in plain text in a file, this is not recommended.

CAUTION Leaving the SSL-enabled servers password in plain text in the servers 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.

To start your SSL-enabled server automatically on UNIX or Linux

1. Using a text editor, open the `start` file.
2. Locate the `-start` line in the script and insert the following:

   `echo "`*password*`"|`

   where *password* is the SSL password you have chosen.

   For example, if the SSL password is `examples`, the edited line might look like this:

   ```
   -start)
     echo "examples"|./$PRODUCT_BIN -d $PRODUCT_SUBDIR/config $@
   ```

Stopping the Proxy Server

This section describes the various methods to stop the Proxy Server on different platforms.

To stop the Proxy Server from the administration interface

1. Access the Server Manager and click the Preferences tab.
2. Click the Start/Stop Server link. The Start/Stop Server page displays.
3. Click the Off button.

The status of the server appears in the Start/Stop Server page.

To stop the Proxy Server on UNIX or Linux

- From the command line, go to *server_root*`/proxy-server`*id* and type `./stop`.

NOTE 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.

- 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 proxy server in `etc/inittab` before shutting down the server; otherwise, the server automatically restarts.

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

NOTE If you have a security module installed with your server, you will be required to enter the appropriate passwords before starting or stopping the server.

To stop the Proxy Server on Windows

- Use Start > Programs > Sun Microsystems > Sun Java System Web Proxy Server *version* > Stop Proxy Server
- From a command prompt, go to *server_root*`\proxy-server`*id* and type `stopsvr.bat` to stop the Proxy Server.
- Use the Sun Java System Proxy Server 4.0 (proxy-server*id*) service in the Services window: Control Panel > Administrative Tools > Services

Restarting the Proxy Server

This section describes the various methods to restart the Proxy Server on different platforms.

Restarting the Server (UNIX or Linux)

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

- Restart it manually.
- Automatically restart it from the `inittab` file

  Note that if you are using a version of UNIX or 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 system reboots.

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.

To restart the Proxy Server from the command line

1. Log in as root if the server runs on ports with numbers lower than 1024; otherwise, log in as root or with the servers user account.
2. At the command-line prompt, type the following line and press Enter:

   server_root`/proxy-server`*id*`/restart`

   where *server_root* is the directory where you installed the server.

- You can use the optional parameter `-i` at the end of the line. 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.

To restart the server using `inittab`

Add the following text on one line in the `/etc/inittab` file:

`prxy:23:respawn:`*server_root*`/proxy-server`*id*`/start -start -i`

where *server_root* is the directory where you installed the server, and `proxy-server`*id* is 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.

To restart the server using System RC Scripts

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

server_root`/proxy-server`*id*`/start`

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

Restarting the Server (Windows)

You can restart the server by

- Using the Services Control Panel.

To restart the server on Windows

1. Use Control Panel > Administrative Tools > Services >
2. Select Sun Java System Web Proxy Server 4.0 (proxy-server*id*) from the list of services.
3. Change the Startup type to Automatic in the Properties window to have your system start the server each time the computer starts or reboots.
4. Click OK.

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 30 seconds. To change the value, add the following line to `magnus.conf` file:

`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.

Viewing Server Settings

During installation, you configure some settings for your Proxy Server. You can view these and other system settings from the Server Manager. The View Server Settings page lists all of the settings for your Proxy Server. This page also tells you if you have unsaved and unapplied changes, in which case you should save the changes and restart the Proxy Server so it can begin using the new configurations.

There are two types of settings, technical and content. The server’s content settings depend on how you have configured your server. Typically, the proxy lists all templates, URL mappings, and access control. For individual templates, this page lists the template name, its regular expression, and the settings for the template such as cache settings.

The proxy server's technical settings come from the `magnus.conf` file and the `server.xml` file, and the content settings come from the `obj.conf` file. These files are located in the server root directory in the subdirectory called `proxy-`*id*`/config`.

To view the settings for the Proxy Server

1. Access the Server Manager and click the Preferences tab.
2. Click the View Server Settings link. The View Server Settings page displays.

Viewing and Restoring Backups of Configuration Files

You can view or restore a backup copy of your configuration files (`server.xml`, `magnus.conf`, `obj.conf`, `mime types`, `server.xml.clfilter`, `magnus.conf.clfilter`, `obj.conf.clfilter`, `socks5.conf`, `bu.conf`, `icp.conf`, `parray.pat`, `parent.pat`, `proxy-id.acl`). This feature lets you go to a previous configuration if you are having trouble with your current configuration. For example, if you several changes to the proxy's configuration and then the proxy does not work the way you thought it should (for example, you denied access to a URL but the proxy will service the request), you can revert to a previous configuration and then redo your configuration changes.

To view a previous configuration

1. Access the Server Manager and click the Preferences tab.
2. Click the Restore Configuration link. The Restore Configuration page displays. The page lists all the previous configurations ordered by date and time.
3. Click the View link to display a listing of the technical and content settings of a particular version.

To restore a backup copy of your configuration files

1. Access the Server Manager and click the Preferences tab.
2. Click the Restore Configuration link. The Restore Configuration page displays. The page lists all the previous configurations ordered by date and time.
3. Click the Restore link for the version you want to restore.

   If you want to restore all files to their state at a particular time, click the Restore to *time* link on the left-most column of the table (*time* being the date and time to which you want to restore).

You can also set the number of backups displayed on the Restore Configuration page.

To set the number of backups displayed

1. Access the Server Manager and click the Preferences tab.
2. Click the Restore Configuration link. The Restore Configuration page displays.
3. In the Set Number Of Sets Of Backups field, enter the number of backups you want to display.
4. Click the Change button.

Configuring System Preferences

The Configure System Preferences page lets you set up or change the basic aspects of your server. The page allows you to change the server user, the number of processes, listen queue size, proxy timeout, and timeout after interrupt for your proxy server. It also allows you to enable DNS, ICP, proxy arrays, and parent arrays.

To modify the system preferences

1. Access the Server Manager and click the Preferences tab.
2. Click the Configure System Preferences link. The Configure System Preferences page displays.
3. Change the options as required, and then click OK.
4. Click Restart Required. The Apply Changes page displays.
5. Click the Restart Proxy Server button to apply the changes.

The options are described in the following sections.

Server User

The Server User is the user account that the proxy uses. The user name you enter as the proxy server user should already exist as a normal user account. When the server starts, it runs as if it were started by this user.

If you want to avoid creating a new user account, you can choose an account used by another server running on the same host, or if you are running a UNIX proxy, you can choose the user `nobody`. However, on some systems the user `nobody` can own files but cannot run programs, which would make it unsuitable as the proxy user name.

On a UNIX system, all the processes that the proxy spawns are assigned to the server user account.

Processes

The Processes field shows how many processes are available to service requests. By default, the value is 1. Do not modify this setting unless required.

Listen Queue Size

The Listen Queue Size field specifies the maximum number of pending connections on a listen socket.

DNS

A Domain Name Service (DNS) restores IP addresses into host names. When a web browser connects to your server, the server gets only the client's IP address, for example, `198.18.251.30`. The server does not have the host name information, such as `www.example.com`. For access logging and access control, the server can resolve the IP address into a host name. On the Configure System Preferences page, you can tell the server whether or not to resolve IP addresses into host names.

ICP

The Internet Cache Protocol (ICP) is a message-passing protocol that enables caches to communicate with one another. Caches can use ICP to send queries and replies about the existence of cached URLs and about the best locations from which to retrieve those URLs. You can enable ICP on the Configure System Preferences page. For more information on ICP, see "Routing through ICP Neighborhoods" on page 279.

Proxy Array

A proxy array is an array of proxies serving as one cache for the purposes of distributed caching. If you enable the proxy array option on the Configure System Preferences page, that means that the proxy server you are configuring is a member of a proxy array, and that all other members in the array are its siblings. For more information on using proxy arrays, see "Routing through Proxy Arrays" on page 289.

Parent Array

A parent array is a proxy array that a proxy or proxy array routes through. So, if a proxy routes through an upstream proxy array before accessing a remote server, the upstream proxy array is considered the parent array. For more information on using parent arrays with your proxy server, see "Routing through Parent Arrays" on page 301.

Proxy Timeout

The proxy timeout is the maximum time between successive network data packets from the remote server before the proxy server times out the request. The default value for proxy timeout is 5 minutes.

NOTE When the remote server uses server-push and the delay between pages is longer than the proxy timeout, the connection could be terminated before the transmission is done. Instead, use client-pull, which sends multiple requests to the proxy.

Tuning the Proxy Server

The Tune Proxy page allows you to change the default parameters to tune your proxy server's performance.

To change the default tuning parameters

1. Access the Server Manager and click the Preferences tab.

2. Click the Tune Proxy link. The Tune Proxy page displays.

3. You may want to modify the width of FTP listings to better suit your needs. Increasing listing width allows longer file names and thus reduces file name truncation. The default width is 80 characters.
4. Click OK.
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Adding and Editing Listen Sockets

Before the server can process a request it must accept the request via a listen socket, then direct the request to the correct server. When you install the Proxy Server one listen socket, ls1, is created automatically. This listen socket uses the IP address `0.0.0.0` and the port number you specified as your proxy server port number during installation. You cannot delete the default listen socket.

Listen sockets are added, edited, and deleted using the Server Manager's Add Listen Socket and Edit Listen Sockets pages.

This section contains the following topics:

- Adding Listen Sockets
- Editing Listen Sockets
- Deleting Listen Sockets

Adding Listen Sockets

To add listen sockets

1. Access the Server Manager and click the Preferences tab.
2. Click the Add Listen Socket link. The Add Listen Socket page displays.
3. Specify the internal name for the listen socket. You cannot change this name after the listen socket has been created.
4. Specify the IP address of the listen socket. Can be in dotted-pair or IPv6 notation. Can also be `0.0.0.0`, `any`, `ANY` or `INADDR_ANY` (all IP addresses).
5. Specify the port number to create the listen socket on. Legal values are `1 - 65535`. On UNIX, creating sockets that listen on ports 1 - 1024 requires superuser privileges. Configure an SSL listen socket to listen on port 443.

6. Specify the server name to be used in the host name section of any URLs the server sends to the client. This affects URLs that the server automatically generates but does not affect the URLs for directories and files stored in the server. This name should be the alias name if your server uses an alias.
7. From the drop-down list, specify whether security should be enabled or disabled for the listen socket.
8. Click OK.
9. Click Restart Required. The Apply Changes page displays.
10. Click the Restart Proxy Server button to apply the changes.

Editing Listen Sockets

To edit listen sockets

1. Access the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link. The Edit Listen Sockets page displays.
3. In the Configured Sockets table, click the link for the listen socket you want to edit. The Edit Listen Sockets page displays.
4. Make the desired changes to the following options:

- **General**
  - **Listen Socket ID.** The internal name for the listen socket. You cannot change this name after a listen socket has been created.
  - **IP Address.** The IP address of the listen socket. This can be in dotted-pair or IPv6 notation. Can also be `0.0.0.0`, `any`, or `ANY` or `INADDR_ANY` (all IP addresses).
  - **Port.** The port number on which to create the listen socket. Legal values are 1-65535. On UNIX, creating sockets that listen on ports 1-1024 requires superuser privileges. Configure an SSL listen socket to listen on port 443.
  - **Server Name.** The default server for this listen socket.
- **Security**

  If security is disabled, only the following parameter is displayed:

  - **Security.** Enables or disables security for the listen socket selected.

  If security is enabled, the following parameters are displayed:

  - **Security.** Enables or disables security for the listen socket selected.
  - **Server Certificate Name.** Select an installed certificate from the drop-down list to use for this listen socket.
  - **Client Authentication.** Specifies whether client authentication is required on this listen socket. This is Optional by default.
  - **SSL Version 2.** Enables or disables SSL Version 2. This is disabled by default.
  - **SSL Version 2 Ciphers.** Lists all ciphers within this suite. Select the ciphers you want to enable for the listen socket you are editing by checking or unchecking the boxes. The default versions will be unchecked.
  - **SSL Version 3.** Enables or disables SSL Version 3. This is enabled by default.
  - **TLS.** Enables or disables TLS, the Transport Layer Security protocol for encrypted communication. This is enabled by default.
  - **TLS Rollback.** Enables or disables TLS Rollback. Note that disabling TLS Rollback leaves connections vulnerable to version rollback attacks. This is enabled by default.
  - **SSL Version 3 and TLS Ciphers.** Lists all ciphers within this suite. Select the ciphers you want to enable for the listen socket you are editing by checking or unchecking the boxes. The default versions will be checked.

- **Advanced**

  - **Number Of Acceptor Threads.** The number of acceptor threads for the listen socket. The recommended value is the number of processors in the machine. The default is 1, legal values are 1-1024.

  Protocol Family. The socket family type. Legal values are `inet`, `inet6`, and `nca`. Use the value `inet6` for IPv6 listen sockets. Specify `nca` to make use of the Solaris™ Network Cache and Accelerator.

5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Deleting Listen Sockets

To delete listen sockets

1. Access the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Select the check box next to the listen socket you want to delete and click OK. You will be prompted to confirm deletion. It is possible to delete any listen socket, provided it is not the only listen socket for that instance.
4. Click Restart Required. The Apply Changes page displays.
5. Click the Restart Proxy Server button to apply the changes.

MIME Types

A Multi-purpose Internet Mail Extension (MIME) type is a standard for multimedia e-mail and messaging. So that you can filter files depending on their MIME type, the proxy server provides a page that lets you create new MIME types for use with your server. The proxy adds the new types to the `mime.types` file. For more information on blocking files based on MIME types, see "Filtering by MIME Type" on page 309.

This section contains the following topics:

- Creating a New MIME Type
- Editing a MIME Type
- Removing a MIME Type

Creating a New MIME Type

To create a MIME type

1. Access the Server Manager, and click the Preferences tab.
2. Click the Create/Edit MIME Types link. The Create/Edit MIME Types page displays showing all the MIME types listed in the proxy's `mime.types` file.

3. Specify the category of the MIME type from the drop-down list. This can be `type`, enc, or `lang`, where `type` is the file or application type, `enc` is the encoding used for compression, and `lang` is the language encoding. For more information on the category, see the online Help.

4. Specify the content type that will appear in the HTTP header.

5. Specify the file suffix. File Suffix refers to the file extensions that map to the MIME type. To specify more than one extension, separate the entries with a comma. The file extensions should be unique. That is, you should not map one file extension to two MIME types.

6. Click the New button to add the MIME type.

Editing a MIME Type

To Edit a MIME type

1. Access the Server Manager, and click the Preferences tab.

2. Click the Create/Edit MIME Types link. The Create/Edit MIME Types page that displays shows all the MIME types listed in the proxy's `mime.types` file.

3. You can edit any MIME type by clicking the Edit link for that MIME type.

4. Make the desired changes and click the Change MIME Type button.

Removing a MIME Type

To Remove a MIME type

1. Access the Server Manager, and click the Preferences tab.

2. Click the Create/Edit MIME Types link. The Create/Edit MIME Types page that displays shows all the MIME types listed in the proxy's `mime.types` file.

3. You can remove any MIME type by clicking the Remove link for that MIME type.

Administering Access Control

The Administer Access Control page allows you to manage access control lists (ACLs). ACLs allow you to control which clients can access your server. ACLs can screen out certain users, groups, or hosts to either allow or deny access to part of your server, and set up authentication so that only valid users and groups can access part of the server. For more information about access control, see Chapter 8, "Controlling Access to Your Server" on page 149.

To manage access control lists

1. Access the Server Manager, and click the Preferences tab.
2. Click the Administer Access Control link. The Administer Access Control page displays.
3. Pick a resource, an existing ACL, or type in the ACL name and click the Edit button. The Access Control Rules for page displays.
4. Make the desired changes and click Submit. For more information about access control see "Setting Access Control for a Server Instance" in Chapter 8, "Controlling Access to Your Server" on page 149.

Configuring the ACL Cache

The Configure ACL Cache page is used to enable or disable the proxy authentication cache, set the proxy authentication cache directory, configure the cache table size, and set the entry expiration time.

To configure the ACL Cache

1. Access the Server Manager and click the Preferences tab.
2. Click the Configure ACL Cache link. The Configure ACL Cache page displays.
3. You can enable or disable the proxy authentication cache.
4. Select the number of users in the user cache from the Proxy Auth User Cache Size drop-down list. The default size is 200.
5. Select the number of group IDs that can be cached for a single UID / cache entry from the Proxy Auth Group Cache Size drop-down list. The default size is 4.

6. Select the number of seconds before cache entries expire. Each time an entry in the cache is referenced, its age is calculated and checked against this value. The entry is not used if its age is greater than or equal to the Proxy Auth Cache Expiration value. 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 Proxy Server when you make changes to the LDAP entries. For example, if this value is set to 120 seconds, the Proxy Server might be out of sync with the LDAP server for as long as 2 minutes. If your LDAP entries are not likely to change often, use a large number. The default expiration value is 2 minutes.

7. Click OK.

8. Click Restart Required. The Apply Changes page displays.

9. Click the Restart Proxy Server button to apply the changes.

Understanding DNS Caching

Proxy Server supports DNS caching to reduce the number of DNS lookups performed by the proxy while it resolves DNS host names into IP addresses.

Configuring the DNS Cache

The Configure DNS Cache page is used to enable or disable DNS caching, set the size of the DNS cache, set the expiration of DNS cache entries, and enable or disable negative DNS caching.

To configure the DNS Cache

1. Access the Server Manager and click the Preferences tab.

2. Click the Configure DNS Cache link. The Configure DNS Cache page displays.

3. You can enable or disable DNS caching.

4. Select the number of entries from the DNS Cache Size drop-down list that can be stored in the DNS cache. The default size is 1024.

5. You can set the DNS cache expiration time. The Proxy Server purges DNS cache entries from the cache when it reaches a pre-set expiration time. The default DNS expiration time is 20 minutes.

6. You can enable or disable caching of errors when the host name is not found.

7. Click OK.

8. Click Restart Required. The Apply Changes page displays.

9. Click the Restart Proxy Server button to apply the changes.

Configuring DNS Subdomains

Some URLs contain host names with many levels of subdomains. It can take the proxy server a long time to do DNS checks if the first DNS server cannot resolve the host name. You can set the number of levels that the Proxy Server will check before returning a “host not found” message to the client.

For example, if the client requests `http://www.sj.ca.example.com/index.html`, it could take a long time for the proxy to resolve that host into an IP address because it might have to go through four DNS servers to get the IP address for the host computer. Because these lookups can take a lot of time, you can configure the proxy server to quit looking up an IP address if the proxy has to use more than a certain number of DNS servers.

To set the levels of subdomains the proxy traverses

1. Access the Server Manager and click the Preferences tab.

2. Click the Configure DNS Subdomains link. The Configure DNS Subdomains page displays.

3. Select a resource from the drop-down list or specify a regular expression.

4. Select the number of levels from the Local Subdomain Depth drop-down list.

5. Click OK.

6. Click Restart Required. The Apply Changes page displays.

7. Click the Restart Proxy Server button to apply the changes.

Configuring HTTP Keep-Alive

The Configure HTTP Client page is used to enable keep-alives on your proxy server.

The proxy supports HTTP keep-alive packets. The proxy, by default, doe s not use keep-alive connections, but for some systems, using the keep-alive feature can improve the proxy's performance. Keep-alives are a TCP/IP feature that keeps a connection open after the request is complete, so that the client can quickly reuse the open connection.

In normal client-server transactions on the web, the client can make several connections to the server that requests multiple documents. For example, if the client requests a web page that has several graphic images, the client needs to make separate requests for each graphic file. Reestablishing connections is time consuming.

To configure HTTP Keep-Alive

1. Access the Server Manager and click the Preferences tab.
2. Click the Configure HTTP Client link. The Configure HTTP Client page displays.
3. Select a resource from the drop-down list. Select a HTTP or HTTPS resource to configure keep-alives on your Proxy Server or specify a regular expression.
4. Specify whether the HTTP client should use persistent connections by clicking the appropriate Keep Alive option.
5. Specify the maximum number of seconds in the Keep Alive Timeout field to keep a persistent connection open. The default value is 29.
6. You can specify whether the HTTP client can reuse existing persistent connections for all types of requests by selecting the appropriate Persistent Connection Reuse option. The default value is off and does not allow persistent connections to be reused for non-GET requests nor for requests with a body.
7. Specify the HTTP protocol version string in the HTTP Version String field. You should not specify this parameter unless you encounter specific protocol interoperability problems.
8. Specify the Proxy Server product name and version in the Proxy Agent Header field.
9. Specify the nickname of the client certificate in the SSL Client Certificate Nickname field to present to the remote server.
10. Select the appropriate SSL Server Certificate Validation option to indicate whether the Proxy Server must validate the certificate presented by the remote server.
11. Click OK.

12. Click Restart Required. The Apply Changes page displays.

13. Click the Restart Proxy Server button to apply the changes.

Chapter 8

Controlling Access to Your Server

This chapter describes how to control access to the Administration Server and the data served by the Proxy Server. Access can be restricted to all data served by the server, or to specific URLs it serves. For example, you can specify that only certain people access specific URLs, or that everyone except those people can see the files. You might allow all clients to access URLs for HTTP, but allow only restricted access to FTP. You could also restrict URLs based on host names or domain names, such as when you have a Proxy Server serving many internal web servers, but want only specific people to access a confidential research project stored on one of those servers.

Before access control can be used on the Administration Server, you must enable distributed administration and configure an administration group in your LDAP database. The information in this chapter is based on the assumption that those tasks have been performed.

This chapter contains the following sections:

What is Access Control?

Access control enables you to determine who can access the Proxy Server, and what parts of the server they can access. You can control access to the entire server or to just parts of the server, such as directories, files, file types, and so on. When an incoming request is evaluated, access is determined based on a hierarchy of rules called access control entries (ACEs). Proxy Server looks for matching entries to determine whether access should be granted or denied. Each ACE specifies whether the server should continue to the next entry in the hierarchy. The collection of ACEs is called an access control list (ACL). When a request is received, the `obj.conf` file is checked 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.

Access is allowed or denied based on:

- Who is making the request (User-Group)
- Where the request is coming from (Host-IP)
- When the request is happening (such as time of day)
- What type of connection is being used (SSL)

This section contains the following topics:

- Access Control for User-Group
- Access Control for Host-IP
- Using Access Control Files
- Configuring the ACL User Cache
- Controlling Access with Client Certificates

Access Control for User-Group

You can limit access to your server to certain users or groups. User-Group access control requires users to enter a user name and password before gaining access to the server. The server compares the information in a client certificate, or the client certificate itself, with a directory server entry.

The Administration Server uses only Basic authentication. To require client authentication on the Administration Server, you must manually edit the ACL files in `obj.conf`, changing the method to SSL.

User-Group authentication is performed by the directory service configured for a server. For more information, see "Configuring Directory Services" on page 49.

The information a directory service uses to implement access control can come from either of the following sources:

- An internal flat file-type database
- An external LDAP database

When the server uses an external LDAP-based directory service, the following types of User-Group authentication methods are supported for server instances:

- Default
- Basic
- SSL
- Digest
- Other

When the server uses an internal file-based directory service, the User-Group authentication methods supported for server instances includes the following:

- Default
- Basic
- Digest

User-Group authentication requires users to authenticate themselves before gaining access. With authentication, users verify their identity by entering a user name and password, by using a client certificate, or with the Digest authentication plug-in. Using client certificates requires encryption.

Default Authentication

Default authentication is the preferred method. The Default setting uses the default method in the `obj.conf` file, or Basic if there is no setting in `obj.conf`. If Default is selected, the ACL rule does not specify a method in the ACL file. Choosing Default enables you to easily change the methods for all ACLs by editing one line in the `obj.conf` file.

Basic Authentication

Basic authentication requires users to enter a user name and password to access the server. It is the default setting. You must create and store a list of users and groups in an LDAP database, such as the Sun Java System Directory Server, or in a file. You must use a directory server installed on a different server root than your Proxy Server, or a directory server installed on a remote computer.

When users attempt to access a resource that has User-Group authentication, users are prompted to enter a user name and password. The server receives this information encrypted or unencrypted, depending on whether encryption is turned on for your server (SSL is enabled).

CAUTION Using Basic authentication without SSL encryption sends the user name and password in unencrypted text across the network. The network packets could be intercepted, and the user name and password could be pirated. Basic authentication is most effective when combined with SSL encryption, Host-IP authentication, or both. Using Digest authentication eliminates this problem.

After authenticating, the user then sees:

- The resource requested (if successfully authenticated)
- A message denying access if the user name or password is invalid

You can customize the message received by unauthorized users. For more information, see "Responding When Access is Denied" on page 171.

SSL Authentication

The server can confirm users' identities with security certificates in two ways:

- Using the information in the client certificate as proof of identity
- Verifying a client certificate published in an LDAP directory (additional)

When the server is configured to use certificate information for authenticating the client, the server:

- Checks to determine if the certificate is from a trusted CA (Certificate Authority). If not, the authentication fails and the transaction ends. To learn how to enable client authentication, see "Setting Security Preferences" on page 90.

- Maps the certificate to a user’s entry using the `certmap.conf` file, if the certificate is from a trusted CA. To learn how to configure the certificate mapping file, see “Using the certmap.conf File” on page 108.
- Checks the ACL rules specified for that user if the certificate maps correctly. Even if the certificate maps correctly, ACL rules can deny access to the user.

Requiring client authentication for controlling access to specific resources differs from requiring client authentication for all connections to the server. If the server is configured to require client authentication for all connections, the client must only present a valid certificate issued by a trusted CA. If the server is configured to use the SSL method for authentication of users and groups, the following must happen:

- The client must present a valid certificate issued by a trusted CA
- The certificate must be mapped to a valid user in LDAP
- The access control list must evaluate properly

When you require client authentication with access control, SSL ciphers must be enabled for your Proxy Server. See Chapter 5, “Using Certificates and Keys” on page 77 for more information about enabling SSL.

To successfully gain access to an SSL-authenticated resource, the client certificate must be from a CA trusted by the Proxy Server. The client certificate must be published in a directory server if the Proxy Server’s `certmap.conf` file is configured to compare the client’s certificate in the browser with the client certificate in the directory server. However, the `certmap.conf` file can be configured to only compare selected information from the certificate to the directory server entry. For example, you could configure `certmap.conf` to only compare the user ID and e-mail address in the browser certificate with the directory server entry. For more information about `certmap.conf` and certificate mapping, see Chapter 5, “Using Certificates and Keys” on page 77. Also see the Proxy Server *Configuration File Reference*.

Digest Authentication

Proxy Server can be configured to perform Digest authentication using either an LDAP-based or a file-based directory service.

Digest authentication allows users to authenticate based on user name and password without sending the user name and password as cleartext. The browser uses the MD5 algorithm to create a digest value using the user's password and some information provided by the Proxy Server.

When the server uses an LDAP-based directory service to perform Digest authentication, this digest value is also computed on the server side using the Digest authentication plug-in, and compared against the digest value provided by the client. If the digest values match, the user is authenticated. For this to work, your directory server must have access to the user's password in cleartext. Sun Java System Directory Server includes a reversible password plug-in using a symmetric encryption algorithm to store data in an encrypted form that can later be decrypted to its original form. Only the Directory Server holds the key to the data.

For LDAP-based Digest authentication, you must enable the reversible password plug-in and the Digest authentication-specific plug-in included with Proxy Server. To configure your Proxy Server to process Digest authentication, set the `digestauth` property of the database definition in the `dbswitch.conf` file, found in *server_root*/`userdb/`.

The server tries to authenticate against the LDAP database based upon the ACL method specified, as shown in Table 8-1. If you do not specify an ACL method, the server uses either Digest or Basic when authentication is required, or Basic if authentication is not required.

The following table lists Digest authentication that is and is not supported by the authentication database.

Table 8-1 Digest Authentication Challenge Generation

ACL Method	Supported by Authentication Database	Not Supported by Authentication Database
Default None specified	Digest and Basic	Basic
Basic	Basic	Basic
Digest	Digest	ERROR

When processing an ACL with `method=digest`, the server attempts to authenticate by:

- Checking for the Authorization request header. If not found, a 401 response is generated with a Digest challenge, and the process stops.
- Checking for the Authorization type. If the Authentication type is Digest, the server then:

- Checks nonce. If not a valid, fresh nonce generated by this server, a 401 response is generated, and the process stops. If stale, a 401 response is generated with `stale=true`, and the process stops.

  The time the nonce remains fresh can be configured by changing the value of the parameter `DigestStaleTimeout` in the `magnus.conf` file, located in *server_root*`/proxy-server_name/config/`. To set the value, add the following line to `magnus.conf`:

  `DigestStaleTimeout` *seconds*

  where *seconds* represents the number of seconds the nonce remains fresh. After the specified seconds elapse, the nonce expires and new authentication is required from the user.

- Checks the realm. If no match, a 401 response is generated, and the process stops.
- Checks existence of the user in the LDAP directory if the authentication directory is LDAP-based, or checks existence of the user in the file database if the authentication directory is file-based. If not found, a 401 response is generated, and the process stops.
- Gets the `request-digest` value from the directory server or file database and checks for a match to the client's `request-digest`. If no match, a 401 response is generated, and the process stops.
- Constructs the `Authorization-Info` header and inserts this into server headers.

Installing the Digest Authentication Plug-in

For Digest authentication using an LDAP-based directory service, you must install the Digest authentication plug-in. This plug-in computes a digest value on the server side, and compares this against the digest value provided by the client. If the digest values match, the user is authenticated.

If you are using a file-based authentication database, you do not need to install the Digest authentication plug-in.

Installing the Digest Authentication Plug-in on UNIX

The Digest authentication plug-in consists of a shared library found in both:

- `libdigest-plugin.lib`
- `libdigest-plugin.ldif`

To install the Digest authentication plug-in on UNIX

1. Make sure this shared library resides on the same server computer on which the Sun Java System Directory Server is installed.

2. Make sure you know the Directory Manager password.

3. Modify the `libdigest-plugin.ldif` file changing all references to `/path/to` to the location where you installed the digest plug-in shared library.

4. To install the plug-in, enter the command:

   ```
   % ldapmodify -D "cn=Directory Manager" -w password -a <
   libdigest-plugin.ldif
   ```

Installing the Digest Authentication Plug-in on Windows

You must copy several `.dll` files from the Proxy Server installation to your Sun Java System Directory Server server computer for the Directory Server to start properly with the Digest plug-in.

To install the Digest authentication plug-in on Windows

1. Access the shared libraries in Proxy Server in:

   server_root`\bin\proxy\bin`

2. Copy the files:

   - `nsldap32v50.dll`
   - `libspnr4.dll`
   - `libplds4.dll`

3. Paste them into either:

   - `\Winnt\system32`
   - The Sun Java System Directory Server install directory: *server_root*`\bin\sldap\server`

Setting the Sun Java System Directory Server to Use the DES Algorithm

The DES algorithm is needed to encrypt the attribute where the digest password is stored.

To set the Directory Server to use the DES algorithm

1. Launch the Sun Java System Directory Server Console.

2. Open your iDS 5.0 instance.

3. Select the Configuration tab.

4. Click the + sign next to plug-ins.
5. Select the DES plug-in.
6. Choose Add to add a new attribute.
7. Enter `iplanetReversiblePassword`.
8. Click Save.
9. Restart your Sun Java System Directory Server instance.

NOTE	To set a Digest authentication password in the `iplanetReversiblePassword` attribute for a user, your entry must include the `iplanetReversiblePasswordobject` object.

Other Authentication

You can create a custom authentication method using the access control API.

Access Control for Host-IP

You can limit access to the Administration Server and its files and directories by making them available only to clients using specific computers. You specify host names or IP addresses for the computers you want to allow or deny. Access to a file or directory using Host-IP authentication appears seamless to the user. Users can access the files and directories immediately, without entering a user name or password.

Because more than one person may use a particular computer, Host-IP authentication is more effective when combined with User-Group authentication. If both methods of authentication are used, a user name and password will be required for access.

Host-IP authentication does not require DNS (Domain Name Service) to be configured on your server. If you choose to use Host-IP authentication, you must have DNS running in your network, and your server must be configured to use it. To enable DNS, access the Server Manager for your server, click the Preferences tab, and then click Configure System Preferences. You will see the DNS settings.

Enabling DNS degrades the performance of Proxy Server because the server is forced to perform DNS lookups. To reduce the effects of DNS lookups 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, specify the following in `obj.conf`:

```
AddLog fn="flex-log" name="access" iponly=1
```

Using Access Control Files

When you use access control on the Administration Server or the files or directories on the server, the settings are stored in a file with the extension `.acl`. Access control files are stored in the directory *server_root*/`httpacl`, with *server_root* being the location where the server is installed. For example, if you installed the server in `/usr/Sun/Servers`, the ACL files for both the Administration Server and each server instance configured on your server would be located in `/usr/Sun/Servers/httpacl/`.

The main ACL file is `generated-proxy-`*serverid*`.acl`. The temporary working file is `genwork-proxy-`*serverid*`.acl`. If you use the Administration Server to configure access, you will have these two files. However, if you want more complex restrictions, you can create multiple files and reference them from the `server.xml` file. A few features are also available only by editing the files, such as restricting access to the server based on the time of day or day of the week.

For more information about access control files and their syntax, see Appendix A, "ACL File Syntax" on page 379. For more information about `server.xml`, see the Proxy Server *Configuration File Reference*.

Configuring the ACL User Cache

By default, Proxy Server caches user and group authentication results in the ACL user cache. You can control the amount of time the ACL user cache is valid by using 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. Setting the value to 0 (zero) turns the cache off. If you use a large number for this value, you may need to restart Proxy Server every time you make changes to the LDAP entries. For example, if this value is set to 120 seconds, Proxy Server might be out of sync with the LDAP directory for as long as two minutes. Only set a large value if your LDAP directory is not likely to change often.

Using the `magnus.conf` parameter of `ACLUserCacheSize`, you can configure the maximum number of entries that can be held in the cache. The default value for this parameter is 200. New entries are added to the head of the list, and entries at the end of this list are recycled to make new entries when the cache reaches its maximum size.

You can also set the maximum number of group memberships that can be cached per user entry using the `magnus.conf` parameter `ACLGroupCacheSize`. The default value for this parameter is 4. Unfortunately, non-membership of a user in a group is not cached, and results in several LDAP directory accesses on every request.

Controlling Access with Client Certificates

If SSL is enabled on your server, client certificates can be used in conjunction with access control. To do this, you must specify that a client certificate is required to access a specific resource. When this feature is enabled on your server, users with a certificate enter their name and password only the first time they attempt to access a restricted resource. Once their identity is established, the server maps their login name and password to that specific certificate. From then on, users no longer need to enter their login name or password when accessing resources for which client authentication is required. When users attempt to access a restricted resource, their client sends the server the client certificate, which the server checks against its list of mappings. If the certificate belongs to a user to whom you have granted access, the resource is served.

NOTE Requiring client authentication for controlling access to specific resources is different than requiring client authentication for all connections to the server. Also, be aware that requiring client certificates for all SSL connections does not automatically map the certificates to users in your databases. To do this, you must specify that a client certificate is required to access a specified resource.

How Access Control Works

When the server receives a request for a page, it uses the rules in the ACL file to determine if access should be granted. The rules can reference the host name or IP address of the computer sending the request. The rules can also reference users and groups stored in the LDAP directory.

The following example shows the possible contents of an ACL file, and provides examples of access control rules.

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to Sun Java System Web Proxy 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 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 cannot gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory “my_stuff”. Furthermore, members
# of GroupB cannot 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
# users 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, and has been added 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");
```

For example, if a user requests the URL:
`http://server_name/my_stuff/web/presentation.html`

Proxy Server first checks access control for the entire server. If the ACL for the entire server is set to continue, the server checks for an ACL for the directory `my_stuff`. If an ACL exists, the server checks the ACEs within the ACL, and then moves on to the next directory. This process continues until an ACL is found that denies access, or until the final ACL for the requested URL (in this case, the file `presentation.html`) is reached.

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.

NOTE If more than one ACL matches, the server uses the last ACL statement that has a match.

Setting Access Control

This section describes the process of restricting access. You can set global access control rules for all servers, and also individually for specific servers. For instance, a human resources department might create ACLs allowing all authenticated users to view their own payroll data, but restrict access for the purpose of updating data to human resources personnel responsible for payroll.

This section contains the following topics:

- Setting Access Control Globally
- Setting Access Control for a Server Instance

NOTE Distributed administration must be configured and activated before global access control can be set.

Setting Access Control Globally

To set access control for all servers

1. Access the Administration Server and click the Global Settings tab.

2. Click the Administer Access Control link.

3. Select the administration server (`proxy-admserv`) from the drop-down list, click Go to load data, and then click New ACL (or Edit ACL).

4. Authenticate if prompted. The Access Control Rules For page displays. The Administration Server has two lines of default access control rules, which cannot be edited.

5. Select Access Control Is On, if not already selected.

6. To add a default ACL rule to the bottom row of the table, click the New Line button. To change the position of an access control restriction, click the up or down arrow.

7. Click Anyone in the Users/Groups column. The User/Group page displays in the lower frame.

8. Select the users and groups to which you will allow access, and click Update. Clicking the List button for Group or User provides lists from which to choose. For more information about the settings, see the online Help. Also see "Specifying Users and Groups" on page 166.

9. Click Anyplace in the From Host column. The From Host page displays in the lower frame.

10. Specify the host names and IP addresses allowed access, and click Update. For more information about the settings, see the online Help. Also see "Specifying the From Host" on page 168.

11. Click All in the Programs column. The Programs page displays in the lower frame.

12. Select the Program Groups or enter the specific file name in the Program Items field to which you will allow access, and click Update. For more information about the settings, see the online Help. Also see "Restricting Access to Programs" on page 169.

13. (Optional) Click the X in the Extra column to add a customized ACL expression. The Customized Expressions page displays in the lower frame. For more information, see "Writing Customized Expressions" on page 170.

14. Select the checkbox in the Continue column, if not already selected. The server evaluates the next line before determining if the user is allowed access. When creating multiple lines, work from the most general restrictions to the most specific.

15. (Optional) Click the trash can icon to delete the corresponding line from the access control rules.

16. (Optional) Click the Response When Denied link to specify the response a user receives when denied access. The Access Deny Response page displays in the lower frame. Select the desired response, specify additional information if appropriate, and then click Update. For more information about the settings, see "Responding When Access is Denied" on page 171.

17. Click Submit to store the new access control rules in the ACL file, or Revert to reset elements in the page to the values they contained before changes were made.

Setting Access Control for a Server Instance

You can create, edit, or delete access control for a specific server instance using the Server Manager. If deleting, do not delete all ACL rules from the ACL files. At least one ACL file containing a minimum of one ACL rule is required to start the server. Deleting all ACL rules and restarting the server will result in a syntax error.

To set access control for a server instance

1. Access the Server Manager for the server instance and click the Preferences tab.

2. Click the Administer Access Control link.

3. Select an ACL using one of the following methods:

   - Select A Resource displays resources that use ACLs to restrict access. Select a resource from the drop-down list, or click Regular Expression to specify a regular expression. For more information, see Chapter 16, "Managing Templates and Resources" on page 349 in the Proxy Server *Administration Guide.*

   - Select An Existing ACL lists all ACLs that are enabled. Existing ACLs that are not enabled do not display in this list. Make your selection from the drop-down list.

- Type In The ACL Name enables you to create named ACLs. Use this option only if you are familiar with ACL files. You must manually edit `obj.conf` if you want to apply named ACLs to resources. For more information, see Appendix A, "ACL File Syntax" on page 379.

4. Click the corresponding Edit button. The Access Control Rules For page displays.

5. Select Access Control Is On, if not already selected.

6. To add a default ACL rule to the bottom row of the table, click the New Line button. To change the position of an access control restriction, click the up or down arrow.

7. To edit the ACL for this server instance, click the action in the Action column. The Allow / Deny page displays in the lower frame.

8. Select Allow if not already selected as the default, and click Update. For more information about Allow or Deny, see "Setting the Action" on page 166.

9. Click Anyone in the Users / Groups column. The User / Group page displays in the lower frame.

10. Select the users and groups to which you will allow access, specify authentication information, and then click Update. Clicking the List button for Group or User provides lists from which to choose. For more information about the settings, see the online Help. Also see "Specifying Users and Groups" on page 166.

11. Click Anyplace in the From Host column. The From Host page displays in the lower frame.

12. Specify the host names and IP addresses allowed access, and click Update. For more information about the settings, see the online Help. Also see "Specifying the From Host" on page 168.

13. Click All in the Rights column. The Access Rights page displays in the lower frame.

14. Specify access rights for this user, and then click Update. For more information, see "Restricting Access to Programs" on page 169.

15. (Optional) Click the X under the Extra column to add a customized ACL expression. The Customized Expressions page displays in the lower frame. For more information, see "Writing Customized Expressions" on page 170.

16. Select the checkbox in the Continue column, if not already selected. The server evaluates the next line before determining if the user is allowed access. When creating multiple lines, work from the most general restrictions to the most specific.

17. (Optional) Click the trash can icon to delete the corresponding line from the access control rules. Do not delete all ACL rules from the ACL files. At least one ACL file containing at least one ACL rule is required to start the server. If you delete all ACL rules in the ACL files and try to restart the server, you will receive a syntax error.

18. (Optional) Click the Response When Denied link to specify the response a user receives when denied access. The Access Deny Response page displays in the lower frame. Select the desired response, specify additional information if appropriate, and then click Update. For more information about the settings, see "Responding When Access is Denied" on page 171.

19. Click Submit to store the new access control rules in the ACL file, or Revert to reset elements in the page to the values they contained before changes were made.

Selecting Access Control Options

The following topics describe the various options you can select when setting access control. For the Administration Server, the first two lines are set as defaults, and cannot be edited.

This section contains the following topics:

- Setting the Action
- Specifying Users and Groups
- Specifying the From Host
- Restricting Access to Programs
- Setting Access Rights
- Writing Customized Expressions
- Turning Access Control Off
- Responding When Access is Denied

Setting the Action

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

- **Allow** means users or systems can access the requested resource
- **Deny** means users or systems cannot access the resource

The server goes through the list of access control entries (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, and if it matches, the next ACE is used. If Continue is not selected, everyone is denied access to the resource. The server continues down the list until it reaches either an ACE that does not match, or one that matches but does not continue. The last matching ACE determines if access is allowed or denied.

Specifying Users and Groups

With user and group authentication, users are prompted to enter a user name and password before they can access the resource specified in the access control rule.

Proxy Server checks lists of users and groups stored either in an LDAP server, such as Sun Java System Directory Server, or in an internal file-based authentication database.

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

The following elements are displayed for Users/Groups on the Access Control Rules For page in the user interface.

- **Anyone (No Authentication)** is the default and means anyone can access the resource without entering a user name 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 for distributed administration can access the pages.
- **Authenticated People Only**
    - **All In The Authentication Database** matches any user who has an entry in the database.

  - **Only The following People** specifies which users and groups to match. You can list users or groups of users individually by separating the entries with commas, or with a wildcard pattern, or you can select from the lists of users and groups stored in the database. **Group** matches all users in the groups you specify. **User** matches the individual users you specify. For the Administration Server, the users must also be in the administrators group you specified for distributed administration.

- **Prompt For Authentication** specifies the message text that displays in the authentication dialog box. You can use this text to describe what the user needs to enter. Depending on the operating system, users see about the first 40 characters of the prompt. Most browsers cache the user name and password and associate them with the prompt text. This means that if the user accesses areas of the server (files and directories) that have the same prompt, the user does not need to retype user names and passwords. Conversely, if you want to force users to reauthenticate for various areas, you must change the prompt for the ACL on that resource.

- **Authentication Methods** specifies the method the server uses for getting authentication information from the client. The Administration Server offers only the Basic method of authentication. The Server Manager offers the following:

  - **Default** uses the default method specified in the `obj.conf` file, or Basic if there is no setting in `obj.conf`. If you select Default, the ACL rule does not specify a method in the ACL file. Choosing Default enables you to 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 user name and password are only encrypted if encryption is turned on for the server (SSL is enabled). Otherwise, names and passwords are sent in cleartext, and can be read if intercepted.

  - **SSL** uses the client certificate to authenticate the user. To use this method, SSL must be turned on for the server. When encryption is on, Basic and SSL methods can be combined.

  - **Digest** uses an authentication mechanism that enables browsers to authenticate users based on user name and password without sending the user name and password as cleartext. The browser uses the MD5 algorithm to create a digest value using the user's password and some information provided by the Proxy Server. This digest value is also computed on the server side using the Digest authentication plug-in and compared against the digest value provided by the client.

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

- **Authentication Database** specifies the database the server will use to authenticate users. This option is only available through the Server Manager. If you choose Default, the server looks for users and groups in a directory service configured as default. If you want to configure individual ACLs to use different databases, select Other, and specify the database. Non-default databases and LDAP directories must be specified in *server_root*/`userdb/dbswitch.conf`. If you use the access control API for a custom database, select Other, and enter the database name.

Specifying the From Host

You can restrict access to the Administration Server based on which computer the request comes from.

The following elements are displayed for From Host on the Access Control Rules For page in the user interface.

- **Anyplace** allows access to all users and systems
- **Only From** enables you to restrict access to specific host names or IP addresses

If the Only From option is selected, enter a wildcard pattern or a comma-separated list in the Host Names or IP Addresses fields. Restricting by host name is more flexible than by IP address. If a user's IP address changes, you do not need to update this list. Restricting by IP address, however, is more reliable. If a DNS lookup fails for a connected client, host name restriction cannot be used.

You can only use the `*` wildcard notation for 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 `*.example.com`. You can set different host names and IP addresses for superusers accessing the Administration Server.

For host names, the `*` must replace an entire component of the name. That is, `*.example.com` is acceptable, but `*users.example.com` is not. When the `*` displays in a host name, it must be the leftmost character. For example, `*.example.com` is acceptable, but `users.*.com` is not.

For the IP address, the `*` must replace an entire byte in the address. For example, `198.95.251.*` is acceptable, but `198.95.251.3*` is not. When the `*` displays in an IP address, it must be the rightmost character. For example, `198.*` is acceptable, but `198.*.251.30`. is not.

Restricting Access to Programs

Access to programs can only be restricted by the Administration Server. Restricting access to programs allows only specified users to view the Server Manager pages, and determines if those users can configure that server. For example, you might allow some administrators to configure the Users and Groups section of the Administration Server, but deny access to the Global Settings section.

You can configure different users to access different functional domains. Once a user is given access to a few selected functional domains, after the user logs in, Administration Server pages from only those functional domains are available to that user.

The following elements are displayed for Programs on the Access Control Rules For page in the user interface.

- **All Programs** allows or denies access to all programs. By default, administrators have access to all programs for a server.
- **Only The Following** enables you to specify the programs to which users have access.
  - **Program Groups** reflect the tabs of the Administration Server, for example, Preferences and Global Settings, and represent access to those pages. When an administrator accesses the Administration Server, the server uses their user name, host, and IP address to determine the pages they can view.
  - **Program Items** enables you to control access to specific pages within a program by entering a page name in the field.

Setting Access Rights

Access rights can only be set by the Server Manager for a server instance. Access rights restrict access to files and directories on your server. In addition to allowing or denying all access rights, you can specify a rule that allows or denies partial access rights. For example, you could allow users read-only access rights to your files so they can view the information but not change the files.

The following elements are displayed for Rights on the Access Control Rules For page in the user interface.

- **All Access Rights** is the default and allows or denies all rights.
- **Only The following Rights** enables you to select a combination of rights to be allowed or denied:

- **Read** permits users to view files, including the HTTP methods GET, HEAD, POST, and INDEX.
- **Write** permits users to change or delete files, including the HTTP methods PUT, DELETE, MKDIR, RMDIR, and MOVE. To delete a file, a user must have both write and delete rights.
- **Execute** permits users to execute server-side applications, such as CGI programs, Java applets, and agents.
- **Delete** permits users who also have write privileges to delete files or directories.
- **List** permits users to access lists of the files in directories that do not contain an `index.html` file.
- **Info** permits users to receive information about the URI, for example `http_head`.

Writing Customized Expressions

You can enter custom expressions for an ACL. Select this option only if you are familiar with the syntax and structure of ACL files. A few features are 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, 8:00 am to 5:00 pm. 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 about valid syntax and ACL files, see Appendix A, “ACL File Syntax” on page 379.

Turning Access Control Off

When you deselect the option labeled Access Control Is On on the Access Control Rules For page, you receive a prompt asking if you want to erase records in the ACL. When you click OK, the ACL entry for that resource is deleted from the ACL file.

If you want to deactivate an ACL, comment out the ACL lines in the file `generated-proxy-`*serverid*`.acl` by using # signs at the start of each line.

From the Administration Server, you could create and turn on access control for a specific server instance and leave it off (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 could not configure the Administration Server.

Responding When Access is Denied

Proxy Server provides a default message when access is denied, and you can customize the response if desired. You can also create a different message for each access control object.

By default, for the Administration Server, users receive the Permission Denied message in *server_root*`/httpacl/admin-denymsg.html`.

To change the Access Denied message

1. Click the Response When Denied link on the Access Control Rules For page.

2. Select the desired response, enter additional information if appropriate (make sure users have access to the response to which they are redirected), and then click Update.

3. Click Submit to save your changes, or Revert to reset the elements in the page to the values they contained before your changes.

Limiting Access to Areas of Your Server

This section describes some commonly used restrictions to a server and its contents. The steps for each procedure detail the specific actions you must take. However, you still must complete the steps described in “Setting Access Control for a Server Instance” on page 163.

This section contains the following topics:

- Restricting Access to the Entire Server
- Restricting Access to a Directory (Path)
- Restricting Access to a File Type
- Restricting Access Based on Time of Day
- Restricting Access Based on Security
- Securing Access to Resources
- Securing Access to Server Instances
- Enabling IP-based Access Control

Restricting Access to the Entire Server

You may want to allow access to users in a group who access the server from computers in a subdomain. For instance, you may have a server for a company department that you only want users to access from computers in a specific subdomain of your network.

To restrict access to the entire server

Using the steps described for setting access control for a server instance (see "Setting Access Control for a Server Instance" on page 163), do the following:

1. Access the Server Manager for the server instance.
2. On the Preferences tab, click the Administer Access Control link.
3. Select the entire server from the drop-down list, click Select, and then click the corresponding Edit button. The Access Control Rules For page displays.
4. Add a new rule to deny access to all.
5. Add another new rule to allow access to a specific group.
6. Use From Host to specify the host names and IP addresses you want to restrict.
7. Click Submit to save your changes.

Restricting Access to a Directory (Path)

You can allow users in a group to read or run applications in directories, and the subdirectories and files, that are controlled by an owner of the group. For example, a project manager might update status information for a project team to review.

To restrict access to directories

Using the steps described for setting access control for a server instance (see "Setting Access Control for a Server Instance" on page 163), do the following:

1. Access the Server Manager for the server instance.
2. On the Preferences tab, click the Administer Access Control link.
3. Select the desired resource from the drop-down list and click Edit.
4. Create a new rule and leave the defaults to deny access to everyone from everywhere.
5. Create another new rule allowing users in a specific group to have read and execute rights only.
6. Create a third new rule to allow a specific user to have all rights.
7. Deselect Continue for the last two rules.
8. Click Submit to save your changes.

Restricting Access to a File Type

You can limit access to file types. For example, you might want to allow only specific users to create programs that run on your server. Anyone would be able to run the programs, but only specified users in the group would be able create or delete them.

To restrict access to file types

Using the steps described for setting access control for a server instance (see "Setting Access Control for a Server Instance" on page 163), do the following:

1. Access the Server Manager for the server instance.
2. On the Preferences tab, click the Administer Access Control link.
3. Click Regular Expression in the Select A Resource section, and specify the regular expression. For example: `*.cgi`.
4. Click Edit.

5. Create a new rule to allow read access to all users.
6. Create another rule that allows write and delete access only to a specified group.
7. Click Submit to save your changes.

For file type restriction, you would leave both Continue boxes selected. If a request for a file comes in, the server then checks the ACL for the file type first.

A `Pathcheck` function is created in `obj.conf` that may include wildcard patterns for files or directories. The entry in the ACL file would appear as follows: `acl"*.cgi";`

Restricting Access Based on Time of Day

You can restrict write and delete access to the server during specified hours or on specified days.

To restrict access based on time of day

Using the steps described for setting access control for a server instance (see “Setting Access Control for a Server Instance” on page 163), do the following:

1. Access the Server Manager for the server instance.
2. On the Preferences tab, click the Administer Access Control link.
3. Select the entire server from the drop-down list in the Select A Resource section, and click Edit.
4. Create a new rule allowing read and execute rights to all. This means that if a user wants to add, update, or delete a file or directory, this rule does not apply and the server searches for another rule that matches.
5. Create another new rule denying write and delete rights to all.
6. Click the X link to create a customized expression.
7. Enter the days of the week and the times of day to be allowed. For example:

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

8. Click Submit to save your changes. Any errors in the custom expression produce an error message. Make corrections and submit again.

Restricting Access Based on Security

You can configure SSL and non-SSL listen sockets for the same server instance. Restricting access based on security allows you to create protection for resources that should only be transmitted over a secure channel.

To restrict access based on security

Using the steps described for setting access control for a server instance (see "Setting Access Control for a Server Instance" on page 163), do the following:

1. Access the Server Manager for the server instance.
2. On the Preferences tab, click the Administer Access Control link.
3. Select the entire server from the drop-down list in the Select A Resource section, and click Edit.
4. Create a new rule allowing read and execute rights to all. This means that if a user wants to add, update, or delete a file or directory, this rule does not apply and the server searches for another rule that matches.
5. Create another new rule denying write and delete rights to all.
6. Click the X link to create a customized expression.
7. Enter `ssl="on"`. For example:

```
user = "anyone" and ssl="on"
```

8. Click Submit to save your changes. Any errors in the custom expression produce an error message. Make corrections and submit again.

Securing Access to Resources

This section describes the additional tasks you must perform to secure access control with Proxy Server, after enabling distributed administration.

This section contains the following topics.

- Securing Access to Server Instances
- Enabling IP-based Access Control

Securing Access to Server Instances

To configure Proxy Server to control access to server instances, edit the *server_root*/httpacl/*.proxy-admserv.acl files to specify the user to whom you want to grant access control privileges. For example:

```
acl "proxy-server_instance";
authenticate (user,group) {
database = "default";
method = "basic";
};
deny absolute (all) user != "UserA";
```

Enabling IP-based Access Control

If the access control entry that refers to the ip attribute is located in the ACL files related to the Administration Server (gen*.proxy-admserv.acl), complete Steps 1 and 2 below.

If the access control entry that refers to the ip attribute is located in the ACL files related to a server instance, complete only Step 1 below for that particular ACL.

To enable IP-based access control

1. Edit the *server_root*/httpacl/gen*.proxy-admserv.acl files to add ip to the authentication list, in addition to user and group, as shown below:

   ```
   acl "proxy-admserv";
   authenticate (user,group,ip) {
   database = "default";
   method = "basic";
   };
   ```

2. Add the following access control entry:

   `deny absolute (all) ip !="`*ip_for_which_access_is_allowed*`";`

   For example:

   ```
   acl "proxy-admserv";
   authenticate (user,group,ip) {
   database = "default";
   method = "basic";
   };
   deny absolute (all) ip !="205.217.243.119";
   ```

Creating ACLs for File-based Authentication

Proxy Server supports the use of file-based authentication databases, which store user and group information in text format in flat files. The ACL framework is designed to work with the file authentication database.

NOTE Proxy Server does not support dynamic flat files. The flat file database is loaded when the server starts up. Any changes to the files come into effect only when the server is restarted.

This section contains the following topics:

- Creating ACLs for Directory Services Based on File Authentication
- Creating ACLs for Directory Services Based on Digest Authentication

An ACL entry can reference a user database using the `database` keyword. For example:

```
acl "default";
   authenticate (user) {
...
   database="myfile";
...
};
```

The *server_root*/`userdb/dbswitch.conf` file contains an entry that defines the file authentication database and its configuration. For example:

```
directory myfiledb file
myfiledb:syntax keyfile
myfiledb:keyfile /path/to/config/keyfile
```

The following table lists the parameters supported by the file authentication database.

Table 8-2 Parameters Supported by the File Authentication Database

Parameter	Description
`syntax`	(Optional) Value is either `keyfile` or `digest`. If not specified, defaults to `keyfile`.
`keyfile`	(Required if `syntax=keyfile`) Path to the file containing user data.
`digestfile`	(Required if `syntax=digest`) Path to the file containing user data for Digest authentication.

CAUTION The maximum length of a line in a file authentication database file is 255. If any line exceeds this limit, the server fails to start and an error is logged in the log file.

NOTE Make sure a file-based authentication directory service is already configured before attempting to set ACLs using a file-based authentication database. For more information, see "Configuring Directory Services" on page 49.

Creating ACLs for Directory Services Based on File Authentication

To create ACLs for directory services based on file authentication

1. Access the Server Manager for the server instance.
2. On the Preferences tab, click the Administer Access Control link.
3. Select the ACL file from the drop-down list, and click Edit.
4. In the Access Control Rules For page, click the Users/Groups link for the ACL entry you want to edit. The User/Group page displays in the lower frame.

5. From the drop-down list under Authentication Database, specify the key file database.

6. Click Update, and then click Submit to save your changes.

When you set an ACL against a key file-based file authentication database, the `dbswitch.conf` file is updated with an ACL entry, such as the sample entry given below:

```
version 3.0;
acl "default";
authenticate (user) {
     prompt = "Sun Java System Proxy Server 4.0";
     database = "mykeyfile";
     method = "basic";
};
deny (all) user = "anyone";
allow (all) user = "all";
```

Creating ACLs for Directory Services Based on Digest Authentication

The file authentication database also supports a file format suitable for use with Digest authentication, per RFC 2617. A hash based on the password and realm is stored. Cleartext passwords are not maintained.

To create ACLs for directory services based on Digest authentication

1. Access the Server Manager for the server instance.

2. On the Preferences tab, click the Administer Access Control link.

3. Select the ACL file from the drop-down list, and click Edit.

4. In the Access Control Rules For page, click the Users/Groups link for the ACL you want to edit. The User/Group page displays in the lower frame.

5. From the drop-down list under Authentication Database, specify the digest database.

6. Click Update, and then click Submit to save your changes.

When you set an ACL against a Digest authentication-based file authentication database, the `dbswitch.conf` file is updated with an ACL entry, such as the sample entry given below:

```
version 3.0;
acl "default";
authenticate (user) {
     prompt = "filerealm";
     database = "mydigestfile";
     method = "digest";
};

deny (all) user = "anyone";
allow (all) user = "all";
```

Chapter 9

Using Log Files

You can monitor your server's activity using several different methods. This chapter discusses how to monitor your server by recording and viewing log files. For information on using the built-performance monitoring services, or SNMP, see Chapter 10, “Monitoring Servers” on page 207.

This chapter contains the following sections:

- About Log Files
- Logging on UNIX and Windows Platforms
- Log Levels
- Archiving Log Files
- Setting Access Log Preferences
- Setting Error Logging Options
- Configuring the LOG Element
- Viewing Access Log Files
- Viewing Error Log Files
- Working with the Log Analyzer
- Viewing Events (Windows)

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 `proxy-server_name/logs/errors` in the server root directory, lists all the errors the server has encountered. The access log, located in `proxy-server_name/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 Proxy 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.

NOTE Due to limitations in the operating system, Proxy Server cannot work with log files larger than 2GB on Linux. As soon as the maximum file size is reached, logging will cease.

Logging on UNIX and Windows Platforms

This section discusses how log files are created. In addition, this section includes the following topics:

- Default Error Logging
- Logging Using syslog
- Logging Using the Windows eventlog

Default Error Logging

On both the UNIX and Windows platforms, logs from the administration server are collected in the administration `proxy-admserv/logs/` directory. Logs from the server instances are collected in the `proxy-server_name/logs/` directory.

The default log level for the entire server can be set.You can redirect `stdout` and `stderr` to the server's event log and direct the log output to the operating system's system log. Additionally, you can direct stdout and stderr content to the server's event log. Log messages by default are sent to stderr in addition to the specified server log file.

Logging Using syslog

For stable operational environments where centralized logging is required syslog is appropriate. For environments where log output is frequently required for diagnostics and debugging, individual server instance logs may be more manageable.

NOTE

- All logged data for the server instance and administration server in one file may prove difficult to read and debug. It is recommended that you use the syslog master log file only for deployed applications that are running smoothly.
- Logged message are intermixed with all other logs from the Solaris daemon applications.

By using the syslog log file, in conjunction with syslogd, and the system log daemon, you can configure the `syslog.conf` file to:

- Log messages to the appropriate system log
- Write messages to the system console
- Forward logged messages to a list of users, or forward logged messages to another syslogd on another host over the network

Since logging to syslog means, logs from Proxy Server, and other daemon applications are collected in the same file, logged messages are enhanced with the following information to identify Proxy Server-specific messages from the particular server instance:

- Unique message ID
- Timestamp
- Instancename
- Program name (proxyd or proxyd-wdog)
- Process ID (PID of the proxyd process)
- Thread ID (optional)
- Server ID

The LOG element can be configured for both the administration server and the server instance in the `server.xml` file.

For more information on the syslog logging mechanism used in the UNIX operating environment, use the following man commands at a terminal prompt:

```
man syslog

man syslogd

man syslog.conf
```

Logging Using the Windows eventlog

For more information on the event log mechanism used in the Windows operating environment, refer to the Windows help system index for the keywords Event Logging.

Log Levels

The following table defines the log levels and messages in Proxy Server, in increasing order of severity.

Table 9-1 Log Levels

Log level	**Description**

`finest`
`finer`
`fine`
	Messages indicate extent of verbosity of debug messages. `finest` gives the maximum verbosity.

`info`	Messages are informative in nature, usually related to server configuration or server status. These messages do not indicate errors that need immediate action.
`warning`	Messages indicate a warning. The message would probably be accompanied by an exception.
`failure`	Messages indicate a failure of considerable importance that can prevent normal application execution.
`config`	Messages relate to a variety of static configuration information, to assist in debugging problems that may be associated with particular configurations.
`security`	Messages indicate a security issue.
`catastrophe`	Messages indicate a fatal error.

Archiving Log Files

You can set up your access and error log files to be automatically archived. At a certain time, or after a specified interval, your logs will be rotated. Proxy 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 Proxy Server saves and names the file "`access.200505160000`," 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 log archive file varies depending upon which type of log rotation you set up.

Proxy 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.<YYYY><MM><DD><HHMM>
```

```
errors.<YYYY><MM><DD><HHMM>
```

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 when you save and apply changes regardless of the present 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.200505172400`. 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 log rotation is enabled, log file rotation starts at server startup. The first 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.

Scheduler-based Log Rotation

This type of log rotation is based on the time and day stored in the `server.xml` file in the *server_root*/`proxy-`*server_name*/`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 scheduler configuration options are stored in `server.xml` in the `server_root/proxy-`*server_name*/`config/` directory. Logs rotated using the scheduler-based method are saved in the following format:

`<`*`original_filename`*`>.<YYYY><MM><DD><HHMM>`

For example, `access` might become `access.200505171630` when it is rotated at 4:30 p.m.

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

Once the rotation starts, Proxy Server creates a new time stamped log file when there is a request or error that needs to be logged to the access or error 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 scheduler-based method, use the Archive Log page in the Server Manager.

Setting Access Log Preferences

During installation, an access log file named `access` is 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.

You can specify logging preferences using the Set Access 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 initialize 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 in `obj.conf`, and `common-log` to record request-specific data in the common log format (used by most HTTP servers).

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.

When changing the format of an existing log file, you should first delete/rename the existing log file OR use a different file name.

To set the access log preferences for the Administration Server

1. Access the Administration Server and click the Preferences tab.
2. Click the Set Access Log Preferences link. The Set Access Log Preferences page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression, and click OK.
4. Specify whether or not to log client accesses. This requires Domain Name Service (DNS) to be enabled.
5. Specify the absolute path for the access log file. As a default, the log files are kept in the logs directory in the server root. If you specify a partial path, the server assumes the path is relative to the logs directory in the server root.

   If you are editing the entire server, the default value for this field is `$accesslog`, the variable that denotes the access log file for the server in the configuration file.
6. Choose whether or not to record domain names or IP addresses of the systems accessing the server in the access log.
7. Choose the type of log file format to use in the access log. The following options are available:

   ❍ **Use Common LogFile Format.** Includes client’s host name, authenticated user name, date and time of request, HTTP header, status code returned to the client, and content length of the document sent to the client.

   ❍ **Only Log.** Allows you to choose which information will be logged. You can choose from the following flexible log format items:

   - **Client Hostname**. The hostname (or IP address if DNS is disabled) of the client requesting access.
   - **Authenticate User Name.** If authentication was necessary, you can have the authenticated user name listed in the access log.
   - **System Date.** The date and time of the client request.
   - **Full Request.** The exact request the client made.

- **Status.** The status code the server returned to the client.
- **Content Length.** The content length, in bytes, of the document sent to the client.
- **HTTP Header, "referer".** The referer specifies the page from which the client accessed the current page. For example, if a user was looking at the results from a text search query, the referer would be the page from which the user accessed the text search engine. Referers allow the server to create a list of backtracked links.
- **HTTP Header, "user-agent".** The user-agent information--which includes the type of browser the client is using, its version, and the operating system on which it is running--comes from the User-agent field in the HTTP header information the client sends to the server.
- **Method.** The HTTP request method used such as GET, PUT, or POST.
- **URI.** Universal Resource Identifier. The location of a resource on the server. For example, for `http://www.a.com:8080/special/docs`, the URI is `special/docs`.
- **Query String Of The URI.** Anything after the question mark in a URI. For example, for `http://www.a.com:8080/special/docs?find_this`, the query string of the URI is `find_this`.
- **Protocol.** The transport protocol and version used.

❍ If you choose a custom format, type it in the Custom Format field.

8. Click OK.
9. Click Restart Required. The Apply Changes page displays.
10. Click the Restart Proxy Server button to apply the changes.

To set the access log preferences for the Server Instance

1. Access the Server Manager and click the Server Status tab.
2. Click the Set Access Log Preferences link. The Set Access Log Preferences page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression, and click OK.
4. Specify whether or not to log client accesses. This requires Domain Name Service (DNS) to be enabled.

5. Specify the absolute path for the access log file. As a default, the log files are kept in the logs directory in the server root. If you specify a partial path, the server assumes the path is relative to the logs directory in the server root.

   If you are editing the entire server, the default value for this field is `$accesslog`, the variable that denotes the access log file for the server in the configuration file.

6. Choose whether or not to record domain names or IP addresses of the systems accessing the server in the access log.

7. Choose the format the log file should be: common, extended, extended-2, only specified information ("Only log" radio button), or custom. If you click Only log, the following flexible log format items are available:

8. Choose the type of log file format to use in the access log. Server access logs can be in Common Logfile Format, Extended Logfile Format, Extended2 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 Proxy Server) what to log. A customizable format uses parameter blocks that you specify to control what gets logged.

   ❍ **Use Common LogFile Format.** Includes client's host name, authenticated user name, date and time of request, HTTP header, status code returned to the client, and content length of the document sent to the client.

   ❍ **Use Extended LogFile Format.** Includes all of the fields of the common log file format as well as some additional fields such as remote status, proxy to client content length, remote to proxy content length, proxy to remote content length, client to proxy header length, proxy to client header length, proxy to remote header length, remote to proxy header length and transfer time.

   ❍ **Use Extended2 LogFile Format.** Includes all of the fields of the extended logfile format as well as some additional fields such as client status, server status, remote status, cache finish status, and actual route.

   ❍ **Only Log.** Allows you to choose which information will be logged. You can choose from the following flexible log format items:

   - **Client Hostname**. The hostname (or IP address if DNS is disabled) of the client requesting access.

   - **Authenticate User Name.** If authentication was necessary, you can have the authenticated user name listed in the access log.

   - **System Date.** The date and time of the client request.

- **Full Request.** The exact request the client made.
- **Status.** The status code the server returned to the client.
- **Content Length.** The content length, in bytes, of the document sent to the client.
- **HTTP Header, "referer".** The referer specifies the page from which the client accessed the current page. For example, if a user was looking at the results from a text search query, the referer would be the page from which the user accessed the text search engine. Referers allow the server to create a list of backtracked links.
- **HTTP Header, "user-agent".** The user-agent information--which includes the type of browser the client is using, its version, and the operating system on which it is running--comes from the User-agent field in the HTTP header information the client sends to the server.
- **Method.** The HTTP request method used such as GET, PUT, or POST.
- **URI.** Universal Resource Identifier. The location of a resource on the server. For example, for `http://www.a.com:8080/special/docs`, the URI is `special/docs`.
- **Query String Of The URI.** Anything after the question mark in a URI. For example, for `http://www.a.com:8080/special/docs?find_this`, the query string of the URI is `find_this`.
- **Protocol.** The transport protocol and version used.
- **Cache Finish Status.**This field specifies whether the cache file was written, refreshed, or returned by an up-to-date check.
- **Remote Server Finish Status.** This field specifies if the request to the remote server was successfully carried out to completion, interrupted by the client clicking the Stop button in Netscape Navigator, or aborted by an error condition.
- **Status Code From Server.** The status code returned from the server.
- **Route To Proxy (PROXY, SOCKS, DIRECT).** The route used to retrieve the resource. The document can be retrieved directly, through a proxy, or through a SOCKS server.
- **Transfer Time.** The length of time of the transfer, in seconds or milliseconds.
- **Header-length From Server Response.** The length of the header from the server response.

  - **Request Header Size From Proxy To Server.** The size of the request header from the proxy to the server.
  - **Response Header Size Sent To Client.** The size of the response header sent to the client.
  - **Request Header Size Received From Client.** The size of the request header received from the client.
  - **Content-length From Proxy To Server Request.** The length, in bytes, of the document sent from the proxy to the server.
  - **Content-length Received From Client.** The length, in bytes, of the document from the client.
  - **Content-length From Server Response.** The length, in bytes, of the document from the server.
  - **Unverified User From Client.** The user name given to the remote server during authentication.

- ❍ If you choose a custom format, type it in the Custom Format field.

9. If you do not want to log client access from certain host names or IP addresses, type them in the host names and IP Addresses fields. Type a wildcard pattern of hosts from which the server should not record accesses. For example, `*.example.com` does not log accesses from people whose domain is example.com. You can type wildcard patterns for host names, IP addresses, or both.
10. Choose whether to include the format string in the log file. If you are using the Proxy Server’s log analyzer, you should include a format string. If you are using a third-party analyzer, you may not want to include a format string in your logfile.
11. Click OK.
12. Click Restart Required. The Apply Changes page appears.
13. Click the Restart Proxy Server button to apply the changes.

Easy Cookie Logging

Proxy 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 `flex-log` 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.

Setting Error Logging Options

Proxy Server allows you to configure the information to be logged in the server's errors logs.

To set the error logging options

1. To set the error logging options from the Administration Server, choose the Preferences tab, and then click the Set Error Log Preferences link.

   To set the error logging options for the server instance from the Server Manager, choose the Server Status tab, and then click the Set Error Log Preferences link.

2. Specify the file that stores messages from the server in the Error Log File Name field.
3. From the Log Level drop-down list, specify the amount of information that should be logged in the errors log. The following options are available:
4. Select the Log Stdout checkbox if you want stdout output to be redirected to the errors log.
5. Select the Log Stderr checkbox if you want stderr output to be redirected to the errors log.
6. Select the Log To Console checkbox to redirect log messages to the console.
7. Select the Use System Logging checkbox if you want to use the UNIX syslog service or Windows Event Logging to produce and manage logs.
8. Click OK.
9. Click Restart Required. The Apply Changes page appears.
10. Click the Restart Proxy Server button to apply the changes.

Configuring the LOG Element

The following table describes the attributes for the LOG element you can configure in the `server.xml` file:

Table 2 `LOG` attribute

Attribute	Default	Description
`file`	`errors`	Specifies the file that stores messages from the server.
`loglevel`	`info`	Controls the default type of messages logged by other elements to the error log. Allowed values are as follows, from highest to lowest: `finest`, `fine`, `fine`, `info`, `warning`, `failure`, `config`, `security`, and `catastrophe`.
`logstdout`	`true`	(optional) If `true`, redirects `stdout` output to the errors log. Legal values are `on`, `off`, `yes`, `no`, `1`, `0`, `true`, `false`.
`logstderr`	`true`	(optional) If `true`, redirects `stderr` output to the errors log. Legal values are `on`, `off`, `yes`, `no`, `1`, `0`, `true`, `false`.
`logtoconsole`	`true`	(optional, UNIX only) If `true`, redirects log messages to the console.
`createconsole`	`false`	(optional, Windows only) If `true`, creates a Windows console for `stderr` output. Legal values are `on`, `off`, `yes`, `no`, `1`, `0`, `true`, `false`.
`usesyslog`	`false`	(optional) If `true`, uses the UNIX `syslog` service or Windows Event Logging to produce and manage logs. Legal values are `on`, `off`, `yes`, `no`, `1`, `0`, `true`, `false`.

Viewing Access Log Files

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 click the View Access Log link.

To view an access log for the server instance from the Server Manager, choose the Server Status tab, and then click the View Access Log link.

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 Access Log Preferences" on page 184 for more information):

```
198.18.17.222 - - [20/May/2005:14:15:49 +0530] "GET http://www.example.com/
HTTP/1.1" 504 622
198.18.17.222 - abc [20/May/2005:14:16:09 +0530] "GET
http://www.test.com/report.zip HTTP/1.1" 504 630
```

Table 9-3 describes the last line of this sample access log.

Table 9-3 The fields in the last line of the sample access log file

Access Log Field	Example
Hostname or IP address of client	198.18.17.222 (In this case, the client's IP address is shown because the proxy server's setting for DNS lookups is disabled;if DNS lookups were enabled, the client's hostname would appear.
RFC 931 information	- (RFC 931 identity not implemented)
Username	abc (username entered by the client for authentication)
Date/time of request	20/May/2005:14:16:09 +0530
Request	GET
Protocol	HTTP/1.1
Status code	504
Bytes transferred	630

Viewing Error Log Files

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 click the View Error Log link.

To view a server instance's error log file, from the Server Manager, choose the Server Status tab, and click the View Error Log link.

The following are three examples of entries in the error log.

```
20/May/2005:14:08:37] info ( 6141): CORE1116: Sun Java System Web Proxy
Server 4.0 B05/10/2005 01:26
20/May/2005:14:08:37] info ( 6142): CORE3274: successful server startup
20/May/2005:14:08:37] security (23246): for host 198.18.148.89 trying to GET
/, deny-service reports: denying service of /
```

Working with the Log Analyzer

The *server_root*/`extras/log_anly` directory contains the log analysis tool that runs through the Server Manager user interface. 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/flexanlg` directory contains the command-line log analyzer for the flexible log file format. However, the Server Manager defaults to using the flexible log file reporting tool, regardless of the log file format you have selected.

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

You must set the library path before attempting to run the `flexanlg` command line utility. The settings for various platforms are as follows:

Solaris and Linux:

`LD_LIBRARY_PATH=`*`server_root`*`/bin/proxy/lib:$LD_LIBRARY_PATH`

AIX:

`LIBPATH=`*`server_root`*`/bin/proxy/lib:$LIBPATH`

HP-UX:

`SHLIB_PATH=`*`server_root`*`/bin/proxy/lib:$SHLIB_PATH`

Windows:

`path=`*`server_root`*`\bin\proxy\bin;%path%`

NOTE Before running the log analyzer, you should archive the server logs. For more information about archiving server logs, see .

You can also type **`./start -shell`** at the command prompt after changing to the `server_root/proxy-server`*id* directory instead of setting the library path.

If you use the extended or extended-2 logging format, the log analyzer generates several reports within the output file in addition to the information that you designate to be reported. The following sections describe these reports.

Transfer Time Distribution Report

The transfer time distribution report shows the time it takes your proxy server to transfer requests. This report displays the information categorized by service time and by percentage finished. The following is a sample transfer time distribution report.

By service time category:

```
< 1 sec [644%] ........................................
< 2 sec [33.3%] ....................
< 3 sec [ 2.7%] .
< 4 sec [ 1.7%] .
< 5 sec [ 0.6%]
< 6 sec [ 0.4%]
< 7 sec [ 0.2%]
< 8 sec [ 0.0%]
< 9 sec [ 0.0%]
```

By percentage finished:

```
< 1 sec [64.4%] ........................................
< 2 sec [97.7%] ....................................
< 3 sec [100.4%]..............................................
```

Status Code Report

The status code report shows which and how many status codes the proxy server received from the remote server and sent to the client. The status code report also provides explanations for all of these status codes. The following is a sample status code report.

```
Code       -From remote-      -To client-    -Explanation-

200        338  [70.7%]       352  [73.6%]    OK

302        33  [ 6.9%]        36  [ 7.5%]     Redirect

304        90  [18.8%]        99  [20.7%]     Not modified

404        3  [ 0.6%]         3  [ 0.6%]      Not found

407                           5  [ 1.0%]      Proxy authorization
                                              required

500                           2  [ 0.4%]      Internal server error

504                           6  [ 1.3%]      Gateway timeout
```

Data Flow Report

The data flow report shows the data flow (the number of bytes transferred) from the client to the proxy, the proxy to the client, the proxy to the remote server, and the remote server to the proxy. For each of these scenarios, the report shows how much data was transferred in the form of headers and content. The data flow report also shows the data flow from the cache to the client. The following is a sample data flow report.

```
                                 Headers      Content      Total

- Client -> Proxy.........       0 MB         0 MB         0 MB

- Proxy  -> Client...........    0 MB         2 MB         3 MB

- Proxy  -> Remote...........    0 MB         0 MB         0 MB

- Remote -> Proxy..........      0 MB         2 MB         2 MB


Approx:

- Cache  -> Client...........    0 MB         0 MB         0 MB
```

Requests and Connections Report

The requests and connections report shows the number of requests the proxy server receives from clients, the number of connections the proxy makes to a remote server (initial retrievals, up-to-date checks, and refreshes), and the number of remote connections the proxy server avoids by using cached documents. The following is a sample requests and connections report.

```
- Total requests............     478

- Remote connections.........     439

- Avoided remote connects....      39 [ 8.2%]
```

Cache Performance Report

The cache performance report shows the performance of the clients' caches, the proxy server's cache, and the direct connections.

Client Cache

NOTE A client cache hit occurs when a client performs an up-to-date check on a document and the remote server returns a 304 message telling the client that the document was not modified. An up-to-date check initiated by a client indicates that the client has its own copy of the document in the cache.

For the client's cache, the report shows:

- **client and proxy cache hits:** a client cache hit in which the proxy server and the client both have a copy of the requested document and the remote server is queried for an up-to-date check with respect to the proxy's copy and the client's request is then evaluated with respect to the proxy's copy. The cache performance report shows the number of requests of this type that the proxy serviced and the average amount of time it took to service these requests.
- **proxy shortcut no-check:** a client cache hit in which the proxy server and the client both have a copy of the requested document and the proxy server tells the client (without checking with the remote server) that the document in the client's cache is up-to-date. The cache performance report shows the number of requests of this type that the proxy serviced and the average time it took to service these requests.

- **client cache hits only:** a client cache hit in which only the client has a cached copy of the requested document. In this type of request, the proxy server directly tunnels the client's If-modified-since GET header. The cache performance report shows the number of requests of this type that the proxy serviced and the average time it took to service these requests.
- **total client cache hits:** the total number of client cache hits and the average amount of time it took to service these requests.

Proxy Cache

A proxy cache hit occurs when a client requests a document from a proxy server and the proxy server already has the document in its cache. For the proxy server's cache hits, the report shows:

- **proxy cache hits with check:** a proxy cache hit in which the proxy server queries the remote server for an up-to-date check on the document. The cache performance report shows the number of requests of this type that the proxy serviced and the average time it took to service these requests.
- **proxy cache hits without check:** a proxy cache hit in which the proxy server does *not* query the remote server for an up-to-date check on the document. The cache performance report shows the number of requests of this type that the proxy serviced and the average time it took to service these requests.
- **pure proxy cache hits:** a proxy cache hit in which the client does not have a cached copy of the requested document. The cache performance report shows the number of requests of this type that the proxy serviced and the average time it took to service these requests.

Proxy Cache Hits Combined

For the proxy cache hits combined, the report shows:

- **total proxy cache hits:** the total number of hits to the proxy server's cache and the average amount of time it took to service these requests.

Direct Transactions

Direct transactions are those that go directly from the remote server to the proxy server to the client without any cache hits. For the direct transactions, the report shows:

- **retrieved documents:** documents retrieved directly from the remote server. The cache performance report shows the number of requests of this type that the proxy serviced, the average time it took to service these requests, and the percentage of total transactions.

- **other transactions:** transactions that are returned with a status code other than 200 or 304. The cache performance report shows the number of requests of this type that the proxy serviced and the average time it took to service these requests.
- **total direct traffic:** requests (both failed requests and successfully retrieved documents) that went directly from the client to the remote server. The cache performance report shows the number of requests of this type that the proxy serviced, the average time it took to service these requests, and the percentage of total transactions.

The following is a sample cache performance report.

```
CLIENT CACHE:
- Client & proxy cache hits... 86 reqs [18.0%] 0.21 sec/req
- Proxy shortcut no-check........ 13 reqs [ 2.7%] 0.00 sec/req
- Client cache hits only.....
- TOTAL client cache hits.......... 99 reqs [20.7%] 0.18 sec/req
PROXY CACHE:
- Proxy cache hits w/check........ 4 reqs [ 0.8%] 0.50 sec/req
- Proxy cache hits w/o check.. 10 reqs [ 2.1%] 0.00 sec/req
- Pure proxy cache hits...... 14 reqs [ 2.9%] 0.14 sec/req
PROXY CACHE HITS COMBINED:
- TOTAL proxy cache hits....... 113 reqs [23.6%] 0.18 sec/req
DIRECT TRANSACTIONS:
- Retrieved documents..313 reqs [65.5%]  0.90 sec/req 2 MB
- Other transactions.. 52 reqs [10.9%] 7.79 sec/req
- TOTAL direct traffic..365 reqs [76.4%] 1.88 sec/req 2 MB
```

Transfer Time Report

The transfer time report shows the information about the time it takes for the proxy server to process a transaction. This report shows values for the following categories:

average transaction time: the average of all transfer times logged.

average transfer time without caching: the average of transfer times for transactions which are not returned from the cache (200 response from remote server).

average with caching, without errors: the average of transfer times for all non-error transactions (2xx and 3xx status codes).

average transfer time improvement: the average transaction time minus the average transfer time with caching, without errors.

The following is a sample transfer time report.

```
- Average transaction time... 1.48 sec/req
- Ave xfer time w/o caching.. 0.90 sec/req
- Ave w/caching, w/o errors.. 0.71 sec/req
- Ave xfer time improvement.. 0.19 sec/req
```

Hourly Activity Report

For each analyzed hour, the hourly activity report shows:

- the load average
- the number of cache hits with no up-to-date check to the remote server
- the number of hits to the proxy server’s cache with an up-to-date check to the remote sever that proves that the document is up-to-date and the document is in the client cache
- the number of hits to the proxy server’s cache with an up-to-date check to the remote sever that proves that the document is up-to-date and the document is *not* in the client cache
- the number of hits to the proxy server’s cache with an up-to-date check to the remote server that caused part of the document to be updated.
- the number of hits to the proxy server’s cache with an up-to-date check to the remote server that returned a new copy of the requested document with a 200 status code.

the number of requests for which documents are directly retrieved from the remote server without any hits to the proxy server’s cache

To run the log analyzer from the Server Manager

1. Access the Server Manager, and click the Server Status tab.
2. Click the Generate Report link. The Generate Report page displays.
3. Type the name of your server; this name appears in the generated report.
4. Choose whether or not the report will appear in HTML or ASCII format.
5. Select the log file you want to analyze.

6. If you want to save the results in a file, type an output filename in the Output File field. If you leave the field blank, the report results print to the screen. For large log files, you should save the results to a file because printing the output to the screen might take a long time.

7. Select whether or not to generate totals for certain server statistics. The following totals can be generated:

    - **Total Hits.** The total number of hits the server received since access logging was enabled.
    - **304 (Not Modified) Status Codes.** The number of times a local copy of the requested document was used, rather than the server returning the page.
    - **302 (Redirects) Status Codes.** The number of times the server redirected to a new URL because the original URL moved.
    - **404 (Not Found) Status Codes.** The number of times the server could not find the requested document or the server did not serve the document because the client was not an authorized user.
    - **500 (Server Error) Status Codes.** The number of times a server-related error occurred.
    - **Total Unique URLs.** The number of unique URLs accessed since access logging was enabled.
    - **Total Unique Hosts.** The number of unique hosts who have accessed the server since access logging was enabled.
    - **Total Kilobytes Transferred.** The number of kilobytes the server transferred since access logging was enabled.

8. Select whether or not to generate general statistics. If you choose to generate statistics, choose from the following:

9. **Find Top *Number* Seconds Of Log.** Generates statistics based on information from the most recent number of seconds.

10. **Find Top *Number* Minutes Of Log.** Generates statistics based on information from the most recent number of minutes.

11. **Find Top *Number* Hours Of Log.** Generates statistics based on information from the most recent number of hours.

12. **Find *Number* Users (If Logged).** Generates statistics based on information from the number of users.

13. **Find Top *Number* Referers (If Logged).** Generates statistics based on information from the number of referers.

14. **Find Top *Number* User Agents (If Logged)**. Generates statistics based on information on the user agents, for example, the browser type, its version, and the operating system.

15. **Find Top *Number* Miscellaneous Logged Items (If Logged).** Generates statistics based on information from the number of user.

16. Select whether or not to generate lists. If you choose to generate lists, specify the items from the following list for which you would like to generate lists:

    - **URLs Accessed.** Displays the URLs that were accessed.

        - ***Number* Most Commonly Accessed URL.** Displays the most commonly accessed URLs or URLs that were accessed more than a specified number of times.

        - **URLs That Were Accessed More Than *Number* Times.** Displays the URL that were accessed more times than the number specified.

    - **Hosts Accessing Your Server.** Displays the hosts that accessed the Proxy Server.

        - ***Number* Hosts Most Often Accessing Your Server.** Displays the hosts most often accessing your server or hosts that have accessed your server more than a specified number of times

        - **Hosts That Accessed Your Server More Than *Number* Times.** Displays the hosts that accessed your server more times than the number specified.

17. Specify the order in which you want to see the results. Prioritize the following items from 1 to 3 in the order that you would like each section to appear in the report. If you chose to not generate any of them, the section will automatically be left out. The options include the following:

    - Find Totals

    - General Statistics

    - Make Lists

18. Click OK. The report appears in a new window.

To run the log analyzer from the command line

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

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

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

Options marked * can be repeated.

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                           Default: none
-o filename: Output log file                          Default: stdout
-m filename: Meta file                                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 time stats -     Default:s5m5h10u10a10r10x10
    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 time stats.
-l [cx,hx]: Make a list of -                          Default: c+3h5
    c(x,+x): Most commonly accessed URL’s
             (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.
```

Viewing Events (Windows)

In addition to logging errors to the server error log, Proxy 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 Proxy Server has a source label of `proxy-server`*id*.

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 the Event Viewer, consult your system documentation.

Chapter 10

Monitoring Servers

This chapter contains information on ways to monitor your server, including the built-in monitoring tool, and Simple Network Management Protocol (SNMP).

You can use SNMP together with Sun Java System 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.

NOTE On Windows, before installing Proxy Server 4 ensure that Windows SNMP components are already installed on your system.

You can view the server's status in real time by using the statistics feature or the SNMP. If you are using UNIX or Linux, you must configure your Proxy Server for SNMP if you plan to use it. This chapter provides the information you need to use SNMP on UNIX or Linux with your Proxy Server.

This chapter contains the following sections:

- Monitoring the Server Using Statistics
- SNMP Basics
- Setting Up SNMP
- Using a Proxy SNMP Agent (UNIX)
- Reconfiguring the SNMP Native Agent
- Installing the SNMP Master Agent
- Enabling and Starting the SNMP Master Agent
- Configuring the SNMP Master Agent
- Enabling the Subagent
- Understanding SNMP Messages

Monitoring the Server Using Statistics

You can use the statistics feature to monitor your server's current activity. The statistics show you 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 large number of requests, you may need to adjust the server configuration or the system's network kernel to accommodate the requests. Statistics are disabled by default because gathering statistics adds overhead to the Proxy Server. Enabling statistics causes the server to begin gathering and saving statistics information.

Once you enable statistics, you can view statistics in the following areas:

- connections
- DNS
- KeepAlive
- cache
- server requests

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.

Processing Proxy Server Statistics

A built-in function called `stats-xml` is used to collect Proxy Server statistics. This function must be enabled to view statistics from the Server Manager or to generate a report using the `perfdump` function. The `stats-xml` function is also used to enable profiling, which is a requirement for monitoring statistics through the use of a custom NSAPI function. Enabling statistics and profiling on the server initializes a server function called `stats-init` in the `obj.conf` file to begin statistics gathering.

```
Init profiling="on" fn="stats-init"
```

It also creates a `NameTrans` directive that allows you to access statistics from a browser window.

```
NameTrans fn="assign-name" name="stats-xml"
from="(/stats-xml|/stats-xml/.*)
```

Finally, enabling statistics adds a `Service` directive to process the `stats-xml` function when the `NameTrans` directive is selected

```
<Object name="stats-xml">
```

```
Service fn="stats-xml"

</Object>
```

NOTE Statistics gathering updates an Init function in the obj.conf. Therefore, you must stop and start your server for these changes to take effect.

You can retrieve the `stats-xml` output using the following URL:

`http://`*computer_name:proxyport*`/stats-xml/proxystats.xml`

This request will return an XML page containing the Proxy Server statistics. Some browsers allow you to view the data within the browser window; others require that you save the data to an external file and view it with an external viewer. The usefulness of this information is not fully apparent without the ability to parse the statistics for different views of the data for analysis. The use of third-party tools can assist in this process. Without a parsing tool, the `stats-xml` output is best observed through the Server Manager or the `perfdump` SAF.

Restricting Access to the stats-xml Output

You should create an ACL for the `/stats-xml` URI if you want to limit the users who can view the `stats-xml` statistics for your server from a browser.

The ACL file must also be referenced in the `stats-xml` object definition in the obj.conf file. For example, if you created a named ACL for the `/stats-xml` URI, you would need to reference the ACL file in a `PathCheck` statement in the object definition as follows:

```
<Object name="stats-xml">

PathCheck fn="check-acl" acl="stats.acl"

Service fn="stats-xml"

</Object>
```

Enabling Statistics

You must activate statistics on Proxy Server before you can monitor performance. This can be done through the Server Manager, or by editing the `obj.conf` and `magnus.conf` files. Users who create automated tools or write customized programs for monitoring and tuning may prefer to work directly with stats-xml.

CAUTION When you enable statistics/profiling, statistics information is made available to any user of your server.

To enable statistics from the Server Manager

1. Access the Server Manager, and click the Server Status tab.
2. Click the Monitor Current Activity link. The Monitor Current Activity page displays.
3. Select the Yes option for the Activate Statistics/Profiling? element to enable statistics.
4. Click OK.
5. Click Restart Required. The Apply Changes page appears.
6. Click the Restart Proxy Server button to apply the changes.

To enable statistics using `stats-xml`

1. Under the default object in obj.conf, add the following line:

```
NameTrans fn="assign-name" name="stats-xml"
from="(/stats-xml|/stats-xml/.*)"
```

2. Add the following Service function to `obj.conf`:

```
<Object name="stats-xml">

Service fn="stats-xml"

</Object>
```

3. Add the `stats-init` SAF to `magnus.conf`.

Here is an example of `stats-init` in `obj.conf`:

```
Init profiling="on" fn="stats-init" update-interval="5"
```

The above example shows you can also designate the following:

- **update-interval.** The period in seconds between statistics updates. A higher setting (less frequent) will be better for performance. The minimum value is 1; the default value is 5.
- **profiling.** Activate NSAPI performance profiling. The default is "no," which results in slightly better server performance. However, if you activate statistics through the user interface, profiling is turned on by default.

Using Statistics

Once you have enabled statistics, you can get a variety of information on how your server instance is running. The statistics are broken up into functional areas.

To access statistics

1. Access the Server Manager, and click the Server Status tab.

2. Click the Monitor Current Activity link.

3. Choose the refresh interval from the Select Refresh Interval drop-down list.

   The refresh interval is the number of seconds between updates of the statistics information displayed.

4. Choose the kind of statistics you want displayed from the Select Statistics To Be Displayed drop-down list. For more information of the types of statistics, see "Displaying Statistics in the Server Manager" on page 209.

5. Click Submit.

   If your server instance is running, and you have enabled Statistics/Profiling, you see a page displaying the kind of statistics you selected. The page is updated every 5-15 seconds, depending upon what you chose for the refresh interval.

6. Select the process ID from the drop-down list.

   You can view current activity through the Server Manager, but these categories are not fully relevant for tuning your server. The perfdump statistics are recommended for tuning your server.

Displaying Statistics in the Server Manager

This section describes how a subset of the `proxystats.xml` data can be viewed in the Server Manager.

You can view totals, maximum values, peak numbers, and bar graphs of information pertaining to connections to the Proxy Server, DNS processing, keep-alive values, cache, and server requests.

The following section describe the types of information that may be obtained for each of these.

Connection Statistics

The following connection statistics are available from the Server Manager:

- Total number of connections
- Maximum number of queued connections
- Peak number of queued connections
- Current number of queued connections
- Number of processes

DNS Statistics

The following DNS statistics are available from the Server Manager:

- Maximum DNS cache entries
- Number of processes
- Number of DNS cache hits (also shown as a bar graph)
- Number of DNS cache misses (also shown as a bar graph)

Keep-Alive Statistics

The following keep-alive statistics are available from the Server Manager:

- Maximum keep-alive connections
- Keep-alive timeout
- Number of processes
- Number of keep-alive hits (also shown as a bar graph)
- Number of keep-alive flushes (also shown as a bar graph)
- Number of keep-alive refusals (also shown as a bar graph)
- Number of keep alive time-outs (also shown as a bar graph)

Cache Statistics

The following file cache statistics are available from the Server Manager:

- Maximum cache age (in seconds)
- Maximum heap cache size
- Maximum memory cache map size
- Number of processes
- Number of cache hits (also shown as a bar graph)
- Number of cache misses (also shown as a bar graph)
- Number of information cache hits (also shown as a bar graph)
- Number of information cache misses (also shown as a bar graph)
- Number of content cache hits (also shown as a bar graph)
- Number of content cache misses (also shown as a bar graph)

Server Request Statistics

The following server statistics are available from the Server Manager.

- Total number of requests
- Number of bytes received
- Number of bytes sent
- Number of processes
- A breakdown of the requests per HTTP server code (also shown as bar graphs). For example, the HTTP server code 200 indicates a fulfilled request.

Monitoring Current Activity Using the perfdump Utility

The perfdump utility is a Server Application Function (SAF) built into Proxy Server that collects various pieces of performance data from the Proxy Server internal statistics and displays them in ASCII text. The perfdump utility allows you to monitor a greater variety of statistics than those available through the Server Manager.

With perfdump, the statistics are unified. Rather than monitoring a single process, statistics are multiplied by the number of processes, which gives you a more accurate view of the server as a whole.

Enabling the perfdump Utility

You can enable the perfdump SAF only after you have enabled the `stats-xml` function and you can only enable it by directly editing the `obj.conf` file.

To enable the perfdump SAF:

1. Add the following object to your `obj.conf` file after the default object:

   ```
   <Object name="perf">

   Service fn="service-dump"

   </Object>
   ```

2. Add the following to the default object:

   ```
   NameTrans fn=assign-name from="/.perf" name="perf"
   ```

3. Restart your server software.

4. Access perfdump by entering this URL:

`http://`*computer_name:proxyport*`/.perf`

You can request the perfdump statistics and specify how frequently (in seconds) the browser should automatically refresh. The following example sets the refresh to every 5 seconds:

`http://`*computer_name:proxyport*`/.perf?refresh=5`

Sample perfdump Output

The following is sample perfdump output:

```
proxyd pid: 6751

Sun Java System Web Proxy Server 4.0 B05/02/2005 15:32 (SunOS DOMESTIC)

Server started Thu May 19 13:15:14 2005
Process 6751 started Thu May 19 13:15:14 2005

ConnectionQueue:
-----------------------------------------
Current/Peak/Limit Queue Length            0/1/4096
Total Connections Queued                   1
Average Queue Length (1, 5, 15 minutes)    0.00, 0.00, 0.00
Average Queueing Delay                     0.09 milliseconds

ListenSocket ls1:
------------------------
Address                 http://0.0.0.0:8081
Acceptor Threads        1

KeepAliveInfo:
--------------------
KeepAliveCount        0/256
KeepAliveHits         0
KeepAliveFlushes      0
KeepAliveRefusals     0
KeepAliveTimeouts     0
KeepAliveTimeout      30 seconds

SessionCreationInfo:
------------------------
Active Sessions         1
Keep-Alive Sessions     0
Total Sessions Created  48/128
```

```
CacheInfo:
------------------
enabled             yes
CacheEntries        0/1024
Hit Ratio           0/0 (  0.00%)
Maximum Age         0

Native pools:
----------------------------
NativePool:
Idle/Peak/Limit               1/1/128
Work Queue Length/Peak/Limit  0/0/0

Server DNS cache disabled

Async DNS disabled

Performance Counters:
------------------------------------------------
                           Average         Total      Percent

Total number of requests:                      1
Request processing time:    0.2559        0.2559

default-bucket (Default bucket)
Number of Requests:                            1    (100.00%)
Number of Invocations:                         7    (100.00%)
Latency:                    0.2483        0.2483    ( 97.04%)
Function Processing Time:   0.0076        0.0076    (  2.96%)
Total Response Time:        0.2559        0.2559    (100.00%)

Sessions:
----------------------------
Process  Status     Function

6751     response   service-dump
```

For more information about these parameters, see “Using Statistics to Tune Your Server” on Chapter 2 of the Sun ONE Web Server 6.1 *Performance Tuning, Sizing, and Scaling Guide* at:

http://docs.sun.com/source/817-6249/index.html

Restricting Access to the perfdump Output

You need to create an ACL for the `/.perf` URI if you want to limit the users who can view the perfdump statistics for your server from a browser.

The ACL file must also be referenced in the perf object definition in the `obj.conf` file. For example, if you created a named ACL for the `/.perf` URI, you would need to reference the ACL file in a `PathCheck` statement in the object definition as follows:

```
<Object name="perf">

PathCheck fn="check-acl" acl="perf.acl"

Service fn="service-dump"

</Object>
```

Using 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 usually 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 requests that caused this function to be called.
- **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 `default-bucket` is predefined by the server. It records statistics for the functions not associated with any user-defined bucket.

Configuration

You must specify all configuration information for performance buckets in the `magnus.conf` and `obj.conf` files. Only the default bucket is automatically enabled.

First, you must enable performance measurement as described in "Monitoring Current Activity Using the perfdump Utility" on page 211.

The following examples show how to define new buckets in `magnus.conf`:

```
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 example above creates three buckets: `acl-bucket`, `file-bucket`, and `cgi-bucket`. To associate these buckets with functions, add `bucket=`*bucket-name* to the `obj.conf` function for which you wish to measure performance.

Example

```
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 returned by `perfdump`.

The report contains the following information:

- `Average`, `Total`, and `Percent` columns give data for each requested statistic.
- `Request Processing Time` is the total time required by the server to process all requests it has received so far.

- `Number of Requests` is the total number of requests for the function.
- `Number of Invocations` is the total number of times 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 of the buckets.
- `Latency` is the time in seconds the Proxy Server takes to prepare for calling the function.
- `Function Processing Time` is the time in seconds Proxy Server spent inside the function. The percentage of `Function Processing Time` and `Total Response Time` is calculated with reference to the total `Request Processing Time`.
- `Total Response Time` is the sum in seconds of `Function Processing Time` and `Latency`.

The following is an example of the performance bucket information available through `perfdump`:

```
Performance Counters:
------------------------------------------------
                           Average         Total      Percent

Total number of requests:                      1
Request processing time:    0.2559        0.2559

default-bucket (Default bucket)
Number of Requests:                            1    (100.00%)
Number of Invocations:                         7    (100.00%)
Latency:                    0.2483        0.2483    ( 97.04%)
Function Processing Time:   0.0076        0.0076    (  2.96%)
Total Response Time:        0.2559        0.2559    (100.00%)
```

SNMP Basics

SNMP is a protocol used to exchange data about network activity. With SNMP, data travels between a managed device and a network management station (NMS). A managed device is anything that runs SNMP: hosts, routers, your proxy server, and other servers on your network. The NMS is a system used to remotely manage that network. Usually, the NMS software will provide a graph to display collected data or use that data to make sure the server is operating within a particular tolerance.

The 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, such as your web servers. 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 proxy server, this information is transferred between the NMS and the server through the use of two types of agents, the subagent and the master agent.

The subagent gathers information about the server and passes the information to the server's master agent. Every server, except for the Administration Server, has as subagent.

NOTE After making any SNMP configuration changes, you must click Apply Required, then restart SNMP subagent.

The master agent communicates with 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, Proxy Server, and the Messaging Server installed on the same host, the subagents for each of the servers would communicate with the same master agent.

Management Information Base

The Proxy Server stores variables pertaining to network management. Variables the master agent can access are called managed objects. These objects are defined in a tree-like structure called the management information base (MIB). The MIB provides access to the Proxy Servers network configuration, status, and statistics. Using SNMP, you can view this information from the NMS. The top level of the MIB tree shows that the internet object identifier has four sub-trees: 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 Sun Java System server MIBs are also located under the enterprises (1) node. Each Sun Java System server subagent provides an MIB for use in SNMP communication. The server reports significant events to the NMS by sending messages or traps containing these variables. The NMS can also query the servers MIB for data, or can remotely change variables in the MIB.Each Sun Java System server has its own MIB. All Sun Java System server MIBs are located at:

`server_root/plugins/snmp`

The Proxy Servers MIB is a file called proxyserv40.mib. This MIB contains the definitions for various variables pertaining to network management for the Proxy Server. You can see administrative information about your Proxy Server and monitor the server in real time using the Proxy Server MIB.

Setting Up SNMP

In general, to use SNMP you must have a master agent and at least one subagent installed and running on 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.

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 are using the AIX platform, your system supports SMUX.)

See your system documentation for information on how to verify this information.

NOTE After changing SNMP settings in the Administration Server, installing a new server, or deleting an existing server, you must perform the following steps:

- (Windows) Restart the Windows SNMP service or reboot the system.
- (UNIX) Restart the SNMP master agent using the Administration Server.

Table 1 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.

If your server meets these conditions....	...follow these procedures. These are discussed in detail in the following sections.
• 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 master agent. 3. Start the proxy SNMP agent. 4. Restart the native agent using a port number other than the master agent port number. 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)

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 Proxy Server master agent. Before you start, be sure to stop the native master agent. (See your system documentation for detailed information.)

NOTE To use a proxy agent, you will need to install it and then start it. You will also have to restart the native SNMP master agent using a port number other than the one the Proxy Server master agent is running on.

This section includes the following topics:

- Installing the Proxy SNMP Agent
- Starting the Proxy SNMP Agent
- Restarting the Native SNMP Daemon

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 221.
2. Install and start the proxy SNMP agent and restart the native SNMP daemon. See "Using a Proxy SNMP Agent (UNIX)" on page 219.
3. Start the SNMP master agent. See "Enabling and Starting the SNMP Master Agent" on page 222.
4. Enable the subagent. See "Enabling the Subagent" on page 227.

eTo 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 of a `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 of a `CONFIG` file, you would enter:

```
# snmpd -P 1161
```

Reconfiguring the SNMP Native Agent

If your SNMP daemon is running on AIX, it supports SMUX. For this reason, you do not 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

To configure the SNMP master agent you must install the Administration Server instance as the `root` user. However, even a non-`root` user can accomplish basic SNMP tasks, such as MIB browsing, on a web server instance by configuring the SNMP sub-agent to work with the master agent.

To install the master SNMP agent

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 Administration Server, choose the Set SNMP Master Agent Trap page from the Global Settings tab.

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 .

7. Type the community string you want to use in the trap. For more information on community strings, see .

8. Click OK.

9. In the Administration Server, choose the Set SNMP Master Agent Community page from the Global Settings tab.

10. Type the community string for the master agent.

11. Choose an operation for the community.

12. Click New.

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.

This section includes the following topics:

- Starting the Master Agent on Another Port
- Manually Configuring the SNMP Master Agent
- Editing the Master Agent CONFIG File

- Defining sysContact and sysLocation Variables
- Configuring the SNMP Subagent
- Starting the SNMP Master Agent

Starting the Master Agent on Another Port

The Administration Interface will not start the SNMP master agent on ports other than 161.

To manually start the master agent on another port

1. Edit `/`*server_root*`/plugins/snmp/magt/CONFIG` to specify the desired port.
2. Run the start script as follows:

   ```
   cd /server_root/proxy-admserv
   ```

   ```
   ./start -shell /server_root/plugins/snmp/magt/magt
   ```

   ```
   /server_root/plugins/snmp/magt/CONFIG
   ```

   ```
   /server_root/plugins/snmp/magt/INIT
   ```

The master agent will then start on the desired port. However, the user interface will be able to detect that the master agent is running.

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

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.

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
987 East Cannon Road
Mountain View, CA 94043
USA”

INITIAL            sysContact “Jill Dawson
email: jdawson@example.com”
```

Configuring the SNMP Subagent

You can configure the SNMP subagent to monitor your server.

To configure the SNMP subagent

1. Access the Server Manager, and click the Server Status tab.
2. Click the Configure SNMP Subagent link. The Configure SNMP Subagent page displays.
3. Enter the name and domain of the server in the Master Host field.
4. Enter the Description of the server, including operating system information.
5. Enter the Organization responsible for the server.
6. Enter the absolute path for the server in the Location field.
7. Enter the name of the person responsible for the server and the person's contact information in the Contact field.
8. Select On to Enable the SNMP Statistics Collection.
9. Click OK.
10. Click Restart Required. The Apply Changes page appears.
11. Click the Restart Proxy Server button to apply the changes.

Starting the SNMP Master Agent

Once you have installed the SNMP master agent, you can start it manually or by using the Administration Server.

Starting the SNMP Master Agent Manually

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` does not 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

1. Log in to the Administration Server.
2. From the Administration Server , choose the Control SNMP Master Agent page from the Global Settings tab.
3. Click Start.

You can also stop and restart the SNMP master agent from the Control SNMP Master Agent page.

Configuring the SNMP Master Agent

Once you have 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 Set SNMP Master Agent Community page in the Administration Server. You also define which SNMP-related operations a particular community can perform. From the Administration Server, 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 Proxy Server. You can also view, edit, and remove the trap destinations you have already configured. When you configure trap destinations using Proxy 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 221. You can use the Server Manager to enable the subagent.

To stop the SNMP function on UNIX or 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 Configure SNMP Subagent page in the Server Manager, and start the subagent from the Control SNMP Subagent 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 Control SNMP Subagent pageor the Services Control Panel for Windows.

NOTE After making any SNMP configuration changes, you must click Apply Required, then restart SNMP subagent.

Understanding SNMP Messages

GET and SET are two types of messages defined by SNMP. GET and SET messages are sent by a network management station (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). These units contain information about variables stored on the managed device, such as the web server. These variables, also known as managed objects, have values and titles that are reported to the NMS as necessary. Protocol data units sent by the server to the NMS are known as "traps." The use of GET, SET, and "trap" messages are illustrated in the following examples.

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.

Part 4

Managing the Proxy Server

Chapter 11

Proxying and Routing URLs

This chapter describes how requests are handled by the proxy server. It also explains how to enable proxying for specific resources and to configure the proxy server to route URLs to different URLs or servers.

This chapter has the following sections:

Enabling/Disabling Proxying for a Resource

You can turn proxying on or off for resources. Resources can be individual URLs, groups of URLs with something in common, or an entire protocol. You can control whether proxying is on for the entire server, for various resources, or for resources as specified in a template file. This means you can deny access to one or more URLs by turning off proxying for that resource. This can be a global way to deny or allow all access to a resource. (You can also allow or deny access to resources by using URL filters. For more information on URL filters, see "Filtering URLs" on page 304.

To enable proxying for a resource

1. Access the Server Manager and click the Routing tab.

2. Click the Enable / Disable Proxying link. The Enable / Disable Proxying page displays.

3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.

4. You can choose a default setting for the resource you specified. You can choose not to proxy that resource (disable proxying), or you can enable proxying of that resource. The following options are available:

- **Use Default Setting Derived From A More General Resource.** The settings for a more general resource that includes this one will be used for this resource.

- **Do Not Proxy This Resource.** This resource cannot be reached through the proxy.

- **Enable Proxying Of This Resource.** The proxy lets clients access this resource (provided they pass the other security and authorization checks). When you enable proxying for a resource, *all methods are enabled*. The read methods, including GET, HEAD, INDEX, POST, and CONNECT for SSL tunneling, and the write methods, including PUT, MKDIR, RMDIR, MOVE, and DELETE, are all enabled for that resource. Barring any other security checks, clients all have read and write access.

5. Click OK.

6. Click Restart Required. The Apply Changes page displays.

7. Click the Restart Proxy Server button to apply the changes.

Routing through Another Proxy

The Set Routing Preferences page is used to configure your proxy server to route certain resources using the derived default configuration or direct connections; or through proxy arrays, ICP neighborhood, another proxy server, or a SOCKS server.

Configuring Routing for a Resource

To configure routing for a resource

1. Access the Server Manager and click the Routing tab.
2. Click the Set Routing Preferences link. The Set Routing Preferences page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression, and clicking OK.
4. Select the radio button for the type of routing you would like for the resource you are configuring. You can choose one of the following:

- **Derived Default Configuration.** The proxy server uses a more general template (that is, one with a shorter, matching regular expression) to determine if it should use the remote server or another proxy. For example, if the proxy routes all `http://.*` requests to another proxy server and all `http://www.*` requests to the remote server, you could create a derived default configuration routing for `http://www.example.*` requests, which would then go directly to the remote server because of the setting for the `http://www.*` template.
- **Direct Connections.** The request will always go directly to the remote server instead of through the proxy.
- **Route Through A SOCKS Server.** The requests for the specified resource will be routed through a SOCKS server. If you choose this option, you need to specify the name (or IP address) and the port number of the SOCKS server that the proxy server will route through.

- **Route Through.** Lets you specify whether you would like to route through a proxy array, ICP neighborhood, parent array, and / or proxy server. If you choose multiple routing methods here, the proxy will follow the hierarchy shown on the form (i.e. proxy array, redirect, ICP, parent array, another proxy). For more information on routing through a proxy server, see "Chaining Proxy Servers" on page 234.

  For information on routing through a SOCKS server, see "Routing through a SOCKS Server" on page 235. For information on routing through proxy arrays, parent arrays, or ICP neighborhoods, see Chapter 12, "Caching" on page 253.

NOTE To enable routing of connect requests on non-default ports (other than 443), you need to change the `ppath` parameter to `connect://.*` in the `obj.conf` file.

5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Chaining Proxy Servers

You can have the proxy access another proxy for some resources instead of accessing the remote server. This means you can chain proxies together. Chaining is a good way to organize several proxies behind a firewall. Chaining also lets you build hierarchical caching.

To route through another proxy server

1. Access the Server Manager and click the Routing tab.
2. Click the Set Routing Preferences link. The Set Routing Preferences page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.
4. Select the Route Through option in the Routing Through Another Proxy section of the page.
5. Select the Another Proxy checkbox.

6. In the Another Proxy field, enter the name or IP address of the proxy sever that you want to route through.

7. Click OK.

8. Click Restart Required. The Apply Changes page displays.

9. Click the Restart Proxy Server button to apply the changes.

Routing through a SOCKS Server

If you already have a remote SOCKS server running on your network, you can configure the proxy to connect to it for specific resources.

To route through a SOCKS server

1. Access the Server Manager and click the Routing tab.

2. Click the Set Routing Preferences link. The Set Routing Preferences page displays.

3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.

4. Select the Route Through option in the Routing Through Another Proxy section of the page.

5. Select the Route Through SOCKS Server option.

6. Specify the name (or IP address) and the port number of the SOCKS server that the proxy server will route through.

7. Click OK.

NOTE Once you have enabling routing through a SOCKS server, you should create proxy routes using the SOCKS v5 Routing page. Proxy routes identify the IP addresses that are accessible through the SOCKS server your proxy routes through. They also specify whether that SOCKS server connects directly to the host.

8. Click Restart Required. The Apply Changes page displays.

9. Click the Restart Proxy Server button to apply the changes.

Forwarding the Client IP Address to the Server

The Forward Client Credentials page is used to configure the proxy to send client credentials to the remote server.

To configure the proxy to send client IP addresses

1. Access the Server Manager and click the Routing tab.

2. Click the Forward Client Credentials link. The Forward Client Credentials page displays.

3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.

4. Change the following options as required:

    ❍ **Client IP Addressing Forwarding.** The Proxy Server does not send the client's IP address to remote servers when making requests for documents. Instead, the proxy acts as the client and sends its IP address to the remote server. However, there are times when you might want to pass on the client's IP address:

    - If your proxy is one in a chain of internal proxies.

    - If your clients need to access servers that depend on knowing the client's IP address. You can use templates to send the client's IP address only to particular servers.

    Select one of the options to configure the proxy to send client IP addresses:

    - **Default.** Enables the Proxy Server to forward the client's IP addresses.

    - **Blocked.** Does not allow the proxy to forward the client's IP addresses.

    - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding IP addresses. The default HTTP header is named `Client-ip`, but you can send the IP address in any header you choose.

    ❍ **Client Proxy Authentication Forwarding.** Select one of the options to configure the proxy to send the client's authentication details:

    - **Default.** Enables the Proxy Server to forward the client's authentication details.

    - **Blocked.** Does not allow the proxy to forward the client's authentication details.

  - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding authentication details.

- **Client Cipher Forwarding**. Select one of the options to configure the proxy to send the name of the client's SSL / TLS cipher suite to remote servers:

  - **Default.** Enables the Proxy Server to forward the name of the client's SSL / TLS cipher suite to remote servers.

  - **Blocked.** Does not allow the proxy to forward the name of the client's SSL / TLS cipher suite to remote servers.

  - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding the name of the client's SSL / TLS cipher suite to remote servers. The default HTTP header is named `Proxy-cipher`, but you can send the name of the client's SSL / TLS cipher suite in any header you choose.

- **Client Keysize Forwarding.** Select one of the options to configure the proxy to send the size of the client's SSL / TLS key to remote servers:

  - **Default.** Enables the Proxy Server to forward the size of the client's SSL / TLS key to remote servers.

  - **Blocked.** Does not allow the proxy to forward the size of the client's SSL / TLS key to remote servers.

  - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding the size of the client's SSL / TLS key to remote servers. The default HTTP header is named `Proxy-keysize`, but you can send the size of the client's SSL / TLS key in any header you choose.

- **Client Secret Keysize Forwarding.** Select one of the options to configure the proxy to send the size of the client's SSL / TLS secret key to remote servers:

  - **Default.** Enables the Proxy Server to forward the size of the client's SSL / TLS secret key to remote servers.

  - **Blocked.** Does not allow the proxy to forward the size of the client's SSL / TLS secret key to remote servers.

  - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding the size of the client's SSL / TLS secret key to remote servers. The default HTTP header is named `Proxy-secret-keysize`, but you can send the size of the client's SSL / TLS secret key in any header you choose.

- **Client SSL Session ID Forwarding.** Select one of the options to configure the proxy to send the client's SSL / TLS session ID to remote servers:
    - **Default.** Enables the Proxy Server to forward the client's SSL / TLS session ID to remote servers.
    - **Blocked.** Does not allow the proxy to forward the client's SSL / TLS session ID to remote servers.
    - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding the client's SSL / TLS session ID to remote servers. The default HTTP header is named `Proxy-ssl-id`, but you can send the client's SSL / TLS session ID in any header you choose.
- **Client Issuer DN Forwarding.** Select one of the options to configure the proxy to send the distinguished name of the issuer of the client's SSL / TLS certificate to remote servers:
    - **Default.** Enables the Proxy Server to forward the distinguished name of the issuer of the client's SSL / TLS certificate to remote servers.
    - **Blocked.** Does not allow the proxy to forward the distinguished name of the issuer of the client's SSL / TLS certificate to remote servers.
    - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding the distinguished name of the issuer of the client's SSL / TLS certificate to remote servers. The default HTTP header is named `Proxy-issuer-dn`, but you can send the name of the issuer of the client's SSL / TLS certificate in any header you choose.
- **Client User DN Forwarding.** Select one of the options to configure the proxy to send the distinguished name of the subject of the client's SSL / TLS certificate to remote servers:
    - **Default.** Enables the Proxy Server to forward the distinguished name of the subject of the client's SSL / TLS certificate to remote servers.
    - **Blocked.** Does not allow the proxy to forward the distinguished name of the subject of the client's SSL / TLS certificate to remote servers.
    - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding the distinguished name of the subject of the client's SSL / TLS certificate to remote servers. The default HTTP header is named `Proxy-user-dn`, but you can send the name of the subject of the client's SSL / TLS certificate in any header you choose.

- **Client SSL/TLS Certificate Forwarding.** Select one of the options to configure the proxy to send the client's SSL / TLS certificate to remote servers:
    - **Default.** Enables the Proxy Server to forward the client's SSL / TLS certificate to remote servers.
    - **Blocked.** Does not allow the proxy to forward the client's SSL / TLS certificate to remote servers.
    - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding the client's SSL / TLS certificate to remote servers. The default HTTP header is named `Proxy-auth-cert`, but you can send the client's SSL / TLS certificate in any header you choose.
- **Client Cache Information Forwarding.** Select one of the options to configure the proxy to send information about local cache hits to remote servers:
    - **Default.** Enables the Proxy Server to forward the information about local cache hits to remote servers.
    - **Blocked.** Does not allow the proxy to forward the information about local cache hits to remote servers.
    - **Enabled Using HTTP Header.** You can specify an HTTP header for the proxy to use when forwarding information about local cache hits to remote servers. The default HTTP header is named `Cache-info`, but you can send the information about local cache hits in any header you choose.
- **Set Basic Authentication Credentials.** Select one of the options to configure the proxy to send a HTTP request:
    - **User.** Specify the user to authenticate.
    - **Password.** Specify the user's password.
    - **Using HTTP Header.** You can specify an HTTP header for the proxy to use to communicate the credentials.

5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Allowing Clients to Check IP Address

To maintain your network's security, your client may have a feature that restricts access to only certain IP addresses. So that your clients can use this feature, the proxy server provides support for checking Java IP Address. This support enables your clients to query the proxy server for the IP address used to retrieve a resource. When this feature is enabled, a client can request that the proxy server send the IP address of the origin server, and the proxy server will attach the IP address in a header. Once the client knows the IP address of the origin server, it can explicitly specify that the same IP address be used for future connections.

To check the Java IP address

1. Access the Server Manager and click the Routing tab.
2. Click the Check Java IP Address link. The Check Java IP Address page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.
4. Select the radio button to either enable, disable or use the default configuration for Java IP address checking.

NOTE The default option uses a derived default configuration from a more general template (that is, one with a shorter, matching regular expression) to determine whether Java IP address checking should be enabled or disabled.

5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Client Autoconfiguration

If your proxy server supports many clients, you can use a client autoconfiguration file to configure all of your browser clients. The autoconfiguration file contains a JavaScript function that determines which proxy, if any, Navigator uses when accessing various URLs. For more information on this feature, see Chapter 17, "Using the Client Autoconfiguration File" on page 355.

Setting the Network Connectivity Mode

You can connect or disconnect the proxy server computer from the network. This feature makes it convenient to install the proxy on a portable computer that you can use for demonstrations.

When the proxy is disconnected from the network, documents are returned directly from the cache—the proxy can't do up-to-date checks, so the documents are retrieved very quickly (the documents might not be up to date; see Chapter 12, "Caching" on page 253 for more information on caching).

Also, if you are not connected to a network, connections never hang because the proxy server is aware that there is no network and never tries to connect to a remote server. You can use this no-network setting when the network is down but the proxy server computer is running.

NOTE Keep in mind that running the proxy disconnected from the network means that you will eventually be accessing stale data from the cache. Also, running without the network makes the proxy security features unnecessary.

Proxy Server offers four network connectivity modes:

- Default mode is derived from the configuration of the most general matching object.
- Normal mode is the normal operating mode for the proxy. The proxy retrieves documents from the content server if they are not already in the cache. If they are in the cache, they may be checked against the content server to determine if they are up to date. If a cached file has changed, it is replaced with the current copy.
- Fast-demo mode is intended for giving smooth demonstrations when the network is available. If a document is found in the cache, the content server is not contacted, not even to find out if the document has changed. This mode gets rid of any latency created by waiting for the content server to respond. If a document is not in the cache, it is retrieved from the content server and cached. The fast-demo mode has less latency than the normal mode, but can occasionally return stale data, because once it has a copy of a document, it does not do up-to-date checks on it.

- No-network mode is designed for portable computers during the time they are not connected to the network. The proxy returns the document if it is in the cache or returns an error if it is not. The proxy never tries to contact the content server, which prevents the proxy from hanging and timing out while trying to get a connection that doesn't exist.

To change the running mode for the proxy server

1. Access the Server Manager and click the Routing tab.
2. Click the Set Connectivity Mode link. The Set Connectivity Mode page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.
4. Select the mode you want.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Changing the Default FTP Transfer Mode

FTP has two different ways to establish a data connection between the FTP server and the client (the proxy acts as a client). The two modes are referred to as PASV and PORT mode FTP.

- **Passive Mode (PASV).** The data connection is initiated from the proxy server, and the FTP server accepts the connection. This is safer for the site running the proxy server because it does not have to accept inbound connections.
- **Active Mode (PORT).** The data connection is initiated by the remote FTP server, and the proxy accepts the incoming connection. If the proxy server is within a firewall, the firewall might block the incoming FTP data connection from the FTP server, which means the PORT mode might not work.

Some FTP sites run a firewall, which makes PASV mode non-functional for proxy servers. Because of this, the proxy server can be configured to use the PORT mode FTP. You can turn on PORT mode for the entire server, or you can turn it on only for specific FTP servers.

NOTE Even when PASV mode is on, the proxy server will use PORT mode if the remote FTP server does not support PASV mode.

If the proxy server is behind a firewall that makes the PORT mode FTP non-functional, you cannot enable PORT mode. If default is selected for the resource, the proxy server uses the mode from a more general resource. If none is specified, PASV mode will be used.

To set the FTP mode

1. Access the Server Manager and click the Routing tab.
2. Click the Set FTP Mode link. The Set FTP Mode page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.
4. Select the FTP transfer mode
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Specifying the SOCKS Name Server IP Address

If your proxy is configured to make its outbound connections through a SOCKS server, you may need to explicitly specify the IP address for the name server to be used with SOCKS.

You should specify the name server IP address if you are resolving outside host names with a DNS server other than an internal DNS service that is inside the firewall.

To specify the SOCKS name server IP address

1. Access the Server Manager and click the Routing tab.
2. Click the Set SOCKS Name Server link. The Set SOCKS Name Server page displays.
3. Enter the IP address of the DNS name server in text field.

4. Click OK.

NOTE The feature that allows you to specify the SOCKS name server IP address used to only be accessible via the SOCKS_NS environment variable. If you set the environment variable and use the SOCKS Name Server Setting form to specify the name server IP address, the proxy will use the IP address specified on the form instead of the environment variable.

5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Configuring HTTP Request Load Balancing

The Configure HTTP Request Load Balancing page is used to distribute the load among the specified origin server.

To configure HTTP request load balancing

1. Access the Server Manager and click the Routing tab.
2. Click the Configure HTTP Request Load Balancing link. The Configure HTTP Request Load Balancing page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.
4. Specify the URL of an origin server in the Server field. If multiple server parameters are given, the Proxy Server will distribute the load among the specified origin server.
5. Specify the name of the cookie in the Sticky Cookie field that when present in a response will cause subsequent requests to stick to that origin server. The default value is `JSESSIONID`.
6. Specify the name of a URI parameter in the Sticky Parameter field to inspect for route information. When the URI parameter is present in a request URI and its value contains a colon, followed by a route ID, the request will “stick” to the origin server identified by that route ID. The default value is `jsessionid.`
7. Specify the name of the HTTP request header in the Route Header field that is used to communicate route IDs to origin servers. The default value is `proxy-jroute`.

8. Specify the name of the cookie in the Route Cookie field that is generated by the Proxy Server when it encounters a sticky cookie in a response.The default value is `JROUTE`.

9. Click the appropriate Rewrite Host option to indicate whether the Host HTTP request header is rewritten to match the host specified by the server parameter.

10. Click the appropriate Rewrite Location option to indicate whether Location HTTP response headers that match the server parameter should be rewritten.

11. Click the appropriate Rewrite Content Location option to indicate whether Content-location HTTP response headers that match the server parameter should be rewritten.

12. Select the checkbox to indicate whether the *headername* HTTP response headers that match the server parameter should be rewritten, where *headername* is a user-defined header name. Specify the headername in the Headername field.

13. Click OK.

14. Click Restart Required. The Apply Changes page displays.

15. Click the Restart Proxy Server button to apply the changes.

Managing URLs and URL Mappings

The Server Manager lets you map URLs to another server, sometimes called a mirror server. When a client accesses the proxy with a mirrored URL, the proxy retrieves the requested document from the mirrored server and not from the server specified in the URL. The client is never aware that the request is going to a different server. You can also redirect URLs; in this case, the proxy returns only the redirected URL to the client (and not the document), so the client can then request the new document. Mapping also allows you to map URLs to a file, as in PAC and PAT mappings.

This section contains the following topics:

- Creating URL Mappings
- Viewing, Editing, or Removing Existing URL Mappings
- Redirecting URLs

Creating URL Mappings

To map a URL, you specify a URL prefix and where to map it. The following sections describe the various types of URL mappings. You can create four types of URL mappings:

- Regular mappings map a URL prefix to another URL prefix. For example, you can configure the proxy to go to a specific URL anytime it gets a request that begins `http://www.example.com`.
- Reverse mappings map a redirected URL prefix to another URL prefix. These are used with reverse proxies when the internal server sends a redirected response instead of the document to the proxy. See Chapter 14, "Using a Reverse Proxy" on page 317 for more information.
- Regular expressions map all URLs matching the expression to a single URL. For example, you can map all URLs matching .*job.* to a specific URL (perhaps one that explains why the proxy server won't let a user go to a particular URL).
- Client autoconfiguration maps URLs to a specific `.pac` file stored on the proxy server. For more information on autoconfiguration files, see Chapter 17, "Using the Client Autoconfiguration File" on page 355.
- Proxy array table (PAT) maps URLs to a specific `.pat` file stored on the proxy server. You should only create this type of mapping from a master proxy. For more information on PAT files and proxy arrays, see "Routing through Proxy Arrays" on page 289.

Clients accessing a URL are sent to a different location on the same server or on a different server. This is useful when a resource has moved or when you need to maintain the integrity of relative links when directories are accessed without a trailing slash.

For example, suppose you have a heavily loaded web server called hi.load.com that you want mirrored to another server called mirror.load.com. For URLs that go to the hi.load.com computer, you can configure the proxy server to use the mirror.load.com computer.

The source URL prefix must be unescaped, but in the destination (mirror) URL, only characters that are illegal in HTTP requests need to be escaped.

CAUTION Do not use trailing slashes in the prefixes!

To create a URL mapping

1. Access the Server Manager and click the URLs tab.

2. Click the Create Mapping link. The Create Mapping page displays.

3. Choose the type of mapping you want to create.

   ❍ **Regular Mappings**. Maps a URL prefix to another URL prefix. For example, you can configure the proxy to go to a specific URL any time it gets a request that begins `http://www.example.com`. If you select this option, the following option is displayed in the lower section of the page:

      - **Rewrite Host.** Click the appropriate option to indicate whether the Host HTTP header is rewritten to match the host specified by the `to` parameter.

   ❍ **Reverse Mappings.** Maps a redirected URL prefix to another URL prefix. These are used with reverse proxies when the internal server sends a redirected response instead of the document to the proxy. For more information about reverse proxy see, Chapter 14, “Using a Reverse Proxy” on page 317. If you select this option, the following option is displayed in the lower section of the page:

      - **Rewrite Location.** Click the appropriate option to indicate whether the Location HTTP response header should be rewritten.

      - **Rewrite Content Location.** Click the appropriate option to indicate whether the Content-location HTTP response header should be rewritten.

      - **Rewrite Headername.** Select the check box to indicate whether the *headername* HTTP response header should be rewritten, where *headername* is a user-defined header name.

   ❍ **Regular Expressions.** Map all URLs matching the expression to a single URL. For more information on regular expressions, Chapter 16, “Managing Templates and Resources” on page 349.

   ❍ **Client Autoconfiguration.** Maps URLs to a specific .pac file stored on the Proxy Server. For more information on autoconfiguration files, Chapter 17, “Using the Client Autoconfiguration File” on page 355.

   ❍ **Proxy Array Table (PAT).** Maps URLs to a specific `.pat` file stored on the Proxy Server. You should only create this type of mapping from a master proxy. For more information on PAT files and proxy arrays, see "Routing through Proxy Arrays" in Chapter 12, “Caching” on page 253

4. Type the map source prefix. For regular and reverse mappings, this should be the part of the URL you want to substitute.

   For regular expression mappings, the URL prefix should be a regular expression that for all the URLs you want to match. If you also choose a template for the mapping, the regular expression will work only for the URLs within the template's regular expression.

   For client autoconfiguration mappings and proxy array table mappings, the URL prefix should be the full URL the client accesses.

5. Type a map destination.

   For all mapping types except client autoconfiguration and proxy array table, this should be the full URL to which to map. For client autoconfiguration mappings, this value should be the absolute path to the `.pac` file on the proxy server's hard disk. For proxy array table mappings, this value should be the absolute path to the `.pat` file on the master proxy's local disk.

6. Select the template name from the drop-down list, or leave the value at NONE if you do not want to apply a template.

7. Click OK to create the mapping.

8. Click Restart Required. The Apply Changes page displays.

9. Click the Restart Proxy Server button to apply the changes.

Viewing, Editing, or Removing Existing URL Mappings

To change your existing mappings

1. Access the Server Manager and click the URLs tab.

2. Click the View/Edit Mappings link. The View/Edit Mappings page displays.

3. To edit a mapping, click the Edit link that is present next to it. You can edit the prefix, the mapped URL, and template that are affected by the mapping. Click OK to confirm your changes.

4. To remove a mapping, click the mapping you want to edit, then click the Remove link that is present next to it.

5. Click Restart Required. The Apply Changes page displays.

6. Click the Restart Proxy Server button to apply the changes.

Redirecting URLs

You can configure the proxy server to return a redirected URL to the client instead of getting and returning the document. With redirection, the client is aware that the URL originally requested has been redirected to a different URL. The client usually requests the redirected URL immediately. Netscape Navigator automatically requests the redirected URL—the user does not have to explicitly request the document a second time.

URL redirection is useful when you want to deny access to an area because you can redirect the user to a URL that explains why access was denied.

To redirect one or more URLs

1. Access the Server Manager and click the URLs tab.
2. Click the Redirect URLs ink. The Redirect URLs page displays.
3. Enter a source URL that is a URL prefix.
4. Enter a URL to redirect to. This URL can either be a URL prefix or a fixed URL.

   If you choose to use a URL prefix as the URL to redirect to, select the radio button next to the URL prefix field and enter a URL prefix. If you choose to use a fixed URL, select the radio button next to the Fixed URL field and enter a fixed URL.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Chapter 12

Caching

This chapter describes how Sun Java™ System Web Proxy Server caches documents. It also describes how you can configure the cache by using the online pages.

This chapter contains the following sections:

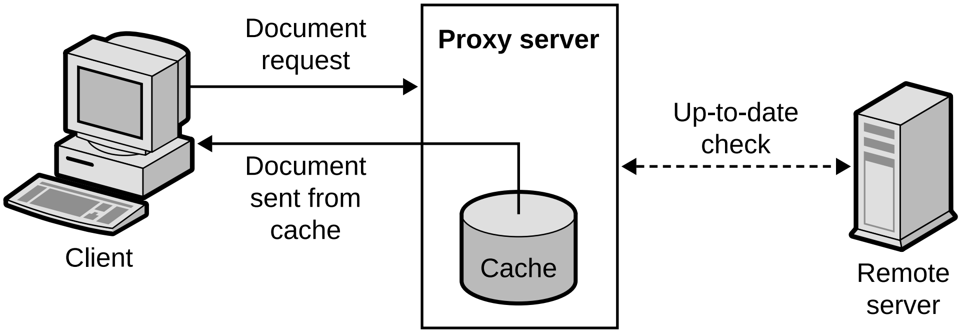
How Caching Works

Caching reduces network traffic and offers faster response time for clients who are using the proxy server instead of going directly to remote servers.

When a client requests a web page or document from the proxy server, the proxy server copies the document from the remote server to its local cache directory structure while sending the document to the client.

When a client requests a document that was previously requested and copied into the proxy cache, the proxy returns the document from the cache instead of retrieving the document from the remote server again (see Figure 12-1). If the proxy determines the file is not up to date, it refreshes the document from the remote server and updates its cache before sending it to the client.

Figure 12-1 Proxy Document Retrieval



Files in the cache are automatically maintained by the Sun Java™ System Web Proxy Server garbage collection utility (CacheGC). The CacheGC automatically cleans the cache on a regular basis to ensure that the cache does not get cluttered with out-of-date documents.

Understanding the Cache Structure

A cache consists of one or more partitions. Conceptually, a partition is a storage area on a disk that you set aside for caching. If you wish to have your cache span several disks, you need to configure at least one cache partition for each disk. Each partition can be independently administered. In other words, you can enable, disable, and configure a partition independently of all other partitions.

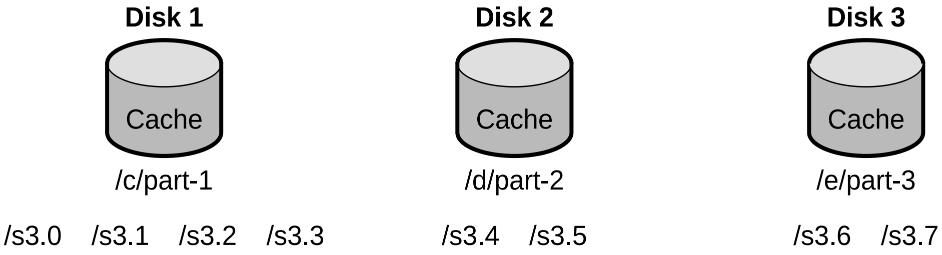
Storing a large number of cached files in a single location can slow performance; therefore, it is a good idea to create several directories, or sections, in each partition. Sections are the next level under partitions in the cache structure. You can have up to 256 sections in your cache across all partitions. The number of cache sections must be a power of 2 (for example, 1, 2, 4, 8, 16, ..., 256).

The final level in the cache structure hierarchy is the subsection. Subsections are directories within sections. Each section has 64 subsections. Cached files are stored in the subsections which is the lowest level in your cache.

Figure 12-2 shows an example cache structure with partitions and sections. In this figure, the cache directory structure divides the total cache into three partitions. The first partition contains four cache sections, and the second two partitions each contain two sections.

Each cache section is noted by s for section, and then a section number. For the section shown as s3.4, the 3 indicates the power of 2 for the number of cache sections (2^3 = 8), and the 4 means the number for the section (for the 8 sections labeled 0 through 7). Therefore, s3.4 means section 5 of 8.

Figure 12-2 Example of a Cache Structure



Distributing Files in the Cache

The Proxy Server uses a specific algorithm to determine the directory where a document should be stored. This algorithm ensures equal distribution of documents in the directories. Equal distribution is important because directories with large numbers of documents tend to cause performance problems.

The Proxy Server uses the RSA MD5 algorithm (Message Digest 5) to reduce the URL to 16 bytes of binary data and uses 8 bytes of this data to calculate a 16-character hexadecimal file name that is used to store the document in the cache.

Setting Cache Specifics

You can enable caching and control which types of protocols your Proxy Server will cache by setting the cache specifics. Cache specifics include the following items:

- Whether your cache is enabled or disabled
- The working directory where it stores its temporary files
- The name of the directory in which you will record the cached URLs
- The size of the cache
- The capacity of the cache
- What types of protocols will be cached
- When to refresh a cached document
- Whether the proxy should track the number of times a document is accessed and report it back to the remote server

To set cache specifics

1. Access the Server Manager, and click the Caching tab.
2. Click the Set Cache Specifics link. The Set Cache Specifics page displays.
3. You can enable or disable the cache by selecting the appropriate option. The cache is enabled by default. For more information, see "Enabling the Cache" on page 256.
4. Enter the working directory. By default the working directory is present under the proxy instance. This can be changed if you want the cache directory in a different location. For more information, see "Creating a Cache Working Directory" on page 256.
5. Click the partition configuration link . The Add/Edit Cache Partitions page displays. You can add a new cache partition or edit existing cache paritions. Cache size is the maximum size the cache is allowed to grow. The maximum cache size is 32GB. For more information, see "Setting Cache Size" on page 256.
6. Click the cache capacity configuration link. The Set Cache Capacity page displays. You can set the cache capacity on the Set Cache Capacity page. For more information, see "Editing Cache Capacity" on page 257.

7. Select the Cache HTTP check box to enable caching of HTTP documents. If you decide that you want your proxy server to cache HTTP documents, you need to determine whether it should always do an up -to-date check for the documents in the cache or if it should check based on an interval. You can also enable or disable the Proxy Server from reporting cache hits to the remote server. For more information, see "Caching HTTP Documents" on page 257. The options include the following:

    - Select the Always Check That The Document Is Up To Date option to ensure that the HTTP document is always up-to- date.
    - Select the number of hours from the Check Only If Last Check More Than drop-down list to specify the refresh interval for the proxy server. The up-to-date check is performed using any one of the following options:
        - **Use Last-modified Factor.** It is the last modified header that is sent by the origin server along with the document.
        - **Use Only Explicit Expiration Information.** The proxy server uses the Expires header to decide if the cache entry is fresh or stale.
    - Select the Never Report Accesses To Remote Server option to prevent the proxy server from reporting the number of accesses to the remote server.
    - Select the Report Cache Hits To Remote Server option to track the number of times a document was accessed and report it back to the remote server.

8. You can set the refresh interval for cached FTP documents. Select the Yes; Reload If Older Than checkbox and also set the time interval by selecting the value from the drop-down list. For more information, see "Caching FTP and Gopher Documents" on page 259.

9. You can set the refresh interval for cached Gopher documents. Select the Yes; Reload If Older Than checkbox and also set the time interval by selecting the value from the drop-down list. For more information, see "Caching FTP and Gopher Documents" on page 259.

10. Click OK.

11. Click Restart required. The Apply Changes page displays.

12. Click the Restart Proxy Server button to apply the changes.

The following sections provide more information about the elements listed on the Set Cache Specifics page and will help you determine which settings will best suit your needs.

Enabling the Cache

Caching is an effective way to reduce network traffic for users of the proxy server. Caching also offers a faster response time for clients by eliminating the need to retrieve a document from a remote server. Your proxy server will function most effectively whenever caching is enabled.

Creating a Cache Working Directory

The cache files are under cache partitions. The working directory you specify on the Set Cache Specifics page is often the parent directory for the cache. All cached files appear in an organized directory structure under the caching directory. If you change the cache directory name or move it to another location, you have to tell the proxy the new location.

You can extend the cache directory structure to multiple file systems so that you can have a large cache structure divided on multiple smaller disks instead of keeping it all on one large disk. Each proxy server must have its own cache directory structure—that is, cache directories cannot be concurrently shared by multiple proxy servers.

Setting Cache Size

The cache size indicates the partition size. Cache size should always be lesser than the cache capacity as it is the maximum size to which the cache can grow. The sum of all the partition sizes must be less than or equal to the cache size.

The amount of disk space available for the proxy cache has a considerable effect on cache performance. If the cache is too small, the Cache GC must remove cached documents to make room on the disk more often, and documents must be retrieved from content servers more often; therefore slowing performance.

Large cache sizes are best because the more cached documents, the less the network traffic load and the faster the response time the proxy provides. Also, the GC removes cached documents if users no longer need them. Barring any file system limitations, cache size can never be too large; the excess space simply remains unused.

You can also have the cache split on multiple disk partitions.

CAUTION Changing the cache structure is time-consuming.

Editing Cache Capacity

You can edit the cache capacity through the Set Cache Specifics page as well as on the Set Cache Capacity page. For more information on editing the cache capacity, see "Setting Cache Capacity" on page 261.

Caching HTTP Documents

Internally, caching HTTP documents differs from caching FTP and Gopher documents. HTTP documents offer caching features that documents of the other protocols do not. However, by setting up and configuring the cache properly, you can ensure that your Proxy Server will cache HTTP, FTP, and Gopher documents effectively.

All HTTP documents have a descriptive header section that the Proxy Server uses to compare and evaluate the document in the proxy cache and the document on the remote server. When the proxy does an up-to-date check on an HTTP document, the proxy sends one request to the server that tells the server to return the document if the version in the cache is out of date. Often, the document has not changed since the last request and therefore is not transferred. This method of checking to see if an HTTP document is up-to-date saves bandwidth and decreases latency.

To reduce transactions with remote servers, the Proxy Server allows you to set a Cache Expiration setting for HTTP documents. The Cache Expiration setting tells the proxy to estimate if the HTTP document needs an up-to-date check before sending the request to the server. The proxy makes this estimate based on the HTTP document's Last-Modified date found in the header.

With HTTP documents, you can also use a Cache Refresh setting. This option specifies whether the proxy always does an up-to-date check (which would override an Expiration setting) or if the proxy waits a specific period of time before doing a check. Table 12-1 shows what the proxy does if both an Expiration setting and a Refresh setting are specified. Using the Refresh setting decreases latency and saves bandwidth considerably.

Table 12-1 Using the Cache Expiration and Cache Refresh settings with HTTP

Refresh setting	Expiration setting	Results
Always do an up-to-date check	(Not applicable)	Always do an up-to-date check

Table 12-1 Using the Cache Expiration and Cache Refresh settings with HTTP

Refresh setting	Expiration setting	Results
User-specified interval	Use document's "expires" header	Do an up-to-date check if interval expired
	Estimate with document's Last-Modified header	Smaller value* of the estimate and expires header

* Using the smaller value guards against getting stale data from the cache for documents that change frequently.

Setting the HTTP Cache Refresh Interval

If you decide that you want your Proxy Server to cache HTTP documents, you need to determine whether it should always do an up-to-date check for documents in the cache or if it should check based on a Cache Refresh setting (up-to-date check interval). For HTTP documents, a reasonable refresh interval would be four to eight hours, for example. The longer the refresh interval, the fewer the number of times the proxy connects with remote servers. Even though the proxy does not do up-to-date checking during the refresh interval, users can force a refresh by clicking the Reload button in the client; this action makes the proxy force an up-to-date check with the remote server.

You can set the refresh interval for HTTP documents on either the Set Cache Specifics page or the Set Caching Configuration page. The Set Cache Specifics page allows you to configure global caching procedures, and the Set Caching Configuration page allows you to control caching procedures for specific URLs and resources.

Setting the HTTP Cache Expiration Policy

You can also set up your server to check if the cached document is up-to-date by using a last-modified factor or explicit expiration information only.

Explicit expiration information is a header found in some HTTP documents that specifies the date and time when that file will become outdated. Not many HTTP documents use explicit Expires headers, so it is better to estimate based on the Last-modified header.

If you decide to have your HTTP documents cached based upon the Last-modified header, you need to select a fraction to use in the expiration estimation. This fraction, known as the LM factor, is multiplied by the interval between the last modification and the time that the last up-to-date check was performed on the document. The resulting number is compared with the time since the last up-to-date check. If the number is smaller than the time interval, the document is not expired. Smaller fractions make the proxy check documents more often. For example, suppose you have a document that was last changed ten days ago. If you set the last-modified factor to 0.1, the proxy interprets the factor to mean that the document is probably going to remain unchanged for one day (10 * 0.1 = 1). The proxy would, in that case, return the document from the cache if the document was checked less than a day ago.

In this same example, if the cache refresh setting for HTTP documents is set to less than one day, the proxy does the up-to-date check more than once a day. The proxy always uses the value (cache refresh or cache expiration) that requires that it update the files more frequently.

You can set the expiration setting for HTTP documents on either the Set Cache Specifics page or the Set Caching Configuration page. The Set Cache Specifics page allows you to configure global caching procedures and the Set Caching Configuration page allows you to control caching procedures, for specific URLs and resources.

Reporting HTTP Accesses to the Remote Server

When a document is cached by Sun Java™ System Web Proxy Server, it can be accessed many times before it is refreshed again. For the remote server, sending one copy to the proxy that will cache it represents only one access, or “hit.” Sun Java™ System Web Proxy Server can count how many times a given document is accessed from the proxy cache between up-to-date checks and then send that hit count back to the remote server in an additional HTTP request header (Cache-Info) the next time the document is refreshed. This way, if the remote server is configured to recognize this type of header, it receives a more accurate account of how many times a document is accessed.

Caching FTP and Gopher Documents

FTP and Gopher do not include a method for checking to see if a document is up-to-date. Therefore, the only way to optimize caching for FTP and Gopher documents is to set a Cache Refresh interval. The Cache Refresh interval is the amount of time the Proxy Server waits before retrieving the latest version of the document from the remote server. If you do not set a Cache Refresh interval, the proxy will retrieve these documents even if the versions in the cache are up-to-date.

Setting FTP and Gopher Cache Refresh Intervals

If you are setting a cache refresh interval for FTP and Gopher, choose one that you consider safe for the documents the proxy gets. For example, if you store information that rarely changes, use a high number (several days). If the data changes constantly, you will want the files to be retrieved at least every few hours. During the refresh time, you risk sending an out-of-date file to the client. If the interval is short enough (a few hours), you eliminate most of this risk while getting noticeably faster response time.

You can set the cache refresh interval for FTP and Gopher documents on either the Set Cache Specifics page or the Set Caching Configuration page. The Set Cache Specifics page allows you to configure global caching procedures, and the Set Caching Configuration page allows you to control caching procedures for specific URLs and resources. For more information on using the Set Cache Specifics page, see "Setting Cache Specifics" on page 254, and for more information on using the Set Caching Configuration page, see "Configuring the Cache" on page 263.

NOTE If your FTP and Gopher documents vary widely (some change often, others rarely), use the Set Caching Configuration page to create a separate template for each kind of document (for example, create a template with resources `ftp://.*.gif`) and then set a refresh interval that is appropriate for that resource.

Creating and Modifying a Cache

Cache partitions are reserved parts of disks or memory that are set aside for caching purposes. If your caching capacity changes, you may want to change or add partitions using the Add/Edit Cache Partitions page. From this page, you can edit a partition's location, mnemonic name, and maximum and minimum sizes. You can also view the cache section table for that partition.

To add cache partitions

1. Access the Server Manager, and click the Caching tab.
2. Click the Add/Edit Cache Partitions link. ThAdd/Edit Cache Partitions page displays.
3. Click the Add Cache Partition button. The Cache Partition Configuration page displays.
4. Enter the appropriate values for the new partition.
5. Click OK.

6. Click Restart Required. The Apply changes page displays.
7. Click Restart Proxy Server button to apply the changes

To modify cache partitions

1. Access the Server Manager, and click the Caching tab.
2. Click the Add/Edit Cache Partitions link. ThAdd/Edit Cache Partitions page displays.
3. Click on the name of the partition that you would like to change.
4. Edit the information.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Setting Cache Capacity

Cache capacity value is used to derive the cache directory structure. The number of sections that can be in the cache directory is derived from the cache capacity. Cache capacity is directly related to the cache hierarchy in the cache directories. The bigger the capacity, the larger the hierarchy. The cache capacity should be equal to or greater than the cache size. Setting the capacity larger than the cache size can be helpful if you know that you plan to increase the cache size later (such as by adding an external disk). The cache capacity can be of maximum 32 GB which will create 256 sections.

To set the cache capacity

1. Access the Server Manager, and click the Caching tab.
2. Click the Set Cache Capacity link. The Set Cache Capacity page displays.
3. Choose a capacity from the New Capacity Range drop-down list.
4. Click OK.
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Managing Cache Sections

The proxy cache is separated into one or more cache sections. You can have up to 256 sections. The number of cache sections must be a power of two (for example, 1, 2, 4, 8, 16, ..., 256). The largest capacity is 32GB (optimum) with 256 cache sections.

If you pick a cache capacity of 500MB, the installer will create 4 cache sections (500 125 = 4); if you choose a cache capacity of 2GB, the installer creates 16 sections (2000 125 = 16). 125MB is chosen as an optimum value for each section to get the number of sections. More the number of sections larger the number of URLs stored and distributed across.

To manage cache sections

1. Access the Server Manager, and click the Caching tab.
2. Click the Manage Sections link. The Manage Sections page displays.
3. Change the information in the table. The sections can be moved among existing partitions.
4. Click OK.
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Setting the Garbage Collection Preferences

The Set Garbage Collection Preferences page is used to set the garbage collection mode.

You can use the cache garbage collector to deletes files from the cache. Garbage collection can be done in either the automatic mode or the explicit mode. The explicit mode is externally scheduled by the administrator using the Schedule Garbage Collection page. Select one of the modes and click OK. Click Restart Required. The Apply Changes page displays. Click the Restart Proxy Server button to apply the changes.

Scheduling Garbage Collection

The Schedule Garbage Collection page allows you to specify the days and time when garbage collection will take place.

To schedule garbage collection:

1. Access the Server Manager, and click the Caching tab.
2. Click the Schedule Garbage Collection link. The Schedule Garbage Collection displays.
3. Select the time at which garbage collection will occur from the Schedule Garbage Collection At list.
4. Specify the day of the week on which garbage collection will occur.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Configuring the Cache

You can configure the kind of caching you want for specific resources, using the Set Caching Configuration page. You can specify several configuration parameter values for URLs matching the regular expression pattern that you specify. This feature gives you fine control of the proxy cache, based on the type of document cached. Configuring the cache can include identifying the following items:

- The cache default
- How to cache pages that require authentication
- How to cache queries
- The minimum and maximum cache file sizes
- When to refresh a cached document
- The cache expiration policy
- The caching behavior for client interruptions
- The caching behavior for failed connections to origin servers

NOTE If you set the cache default for a particular resource to either Derived configuration or Don’t cache, the cache configuration options will not appear on the Set Caching Configuration page. However, if you choose a cache default of Cache for a resource, you can specify several other configuration items.

To configure the cache

1. Access the Server Manager, and click the Caching tab.
2. Click the Set Caching Configuration page. The Set Caching Configuration page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression, and click OK.
4. Change the configuration information.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Caching Configuration Elements

The following sections describe the items listed on the Set Caching Configuration page. These sections include information that will help you to determine which configuration will best suit your needs.

Setting the Cache Default

The proxy server allows you to identify a cache default for specific resources. A resource is a type of file that matches certain criteria that you specify. For instance, you may want your server to automatically cache all documents from the domain company.com. If so, click the Regular Expression button on the top of the Set Caching Configuration page and, in the field that appears, enter

```
[a-z] *://[^/:]\.company\.com.*.
```

By default, the Cache option is selected. Your server automatically caches all cacheable documents from that domain. For more information on regular expressions, see Understanding Regular Expressions.

NOTE	If you set the cache default for a particular resource to either Derived configuration or Don’t cache, it is not necessary to configure the cache for that resource. However, if you choose a cache default of Cache for a resource, you can specify several other configuration items. For a list of these items, see “Configuring the Cache” on page 263.

The cache default for HTTP, FTP, and Gopher can also be set on the Set Cache Specifics page.

Caching Pages That Require Authentication

You can have your server cache files that require user authentication. If you choose to have your Proxy Server cache these files, it tags the files in the cache so that if a user asks for them, it knows that the files require authentication from the remote server.

Because the proxy server does not know how remote servers authenticate and it does not know users' IDs or passwords, it will simply force an up-to-date check with the remote server each time a request is made for a document that requires authentication. The user therefore must enter an ID and password to gain access to the file. If the user has already accessed that server earlier in the Navigator session, Navigator automatically sends the authentication information without prompting the user for it.

If you do not enable the caching of pages that require authentication, the proxy assumes the default, which is to not cache them.

Caching Queries

Cached queries only work with HTTP documents. You can limit the length of queries that are cached, or you can completely inhibit caching of queries. The longer the query, the less likely it is to be repeated, and the less useful it is to cache.

These caching restrictions apply for queries: the access method has to be GET, the document must not be protected (unless caching of authenticated pages is enabled), and the response must have at least a Last-modified header. This requires the query engine to indicate that the query result document can be cached. If the Last-modified header is present, the query engine should support a conditional GET method (with an If-modified-since header) in order to make caching effective; otherwise it should return an Expires header.

Setting Minimum and Maximum Cache File Sizes

You can set the minimum and maximum sizes for files cached by your Proxy Server. You may want to set a minimum size if you have a fast network connection. If your connection is fast, small files may be retrieved so quickly that it is not necessary for the server to cache them. In this instance, you would want to cache only larger files. You may want to set a maximum file size to make sure that large files do not occupy too much of your proxy's disk space.

Setting the Up-to-date Checking Policy

You can use this option to ensure that the HTTP document is always up-to- date. You can also specify the refresh interval for the Proxy Server.

Setting Expiration Policy

You can set the Expiration Policy using the last modified factor or the explicit expiration information.

Setting Cache Behavior for Client Interruptions

If a document is only partly retrieved and the client interrupts the data transfer, the proxy has the ability to finish retrieving the document for the purpose of caching it. The proxy's default is to finish retrieving a document for caching if at least 25 percent of it has already been retrieved. Otherwise, the proxy terminates the remote server connection and removes the partial file. You can raise or lower the client interruption percentage.

Behaviour On Failure To Connect To Server

If an up-to-date check on a stale document fails because the origin server is unreachable, you can specify whether the proxy sends the stale document from the cache.

Caching Local Hosts

If a URL requested from a local host lacks a domain name, the Proxy Server will not cache it in order to avoid duplicate caching. For example, if a user requests `http://machine/filename.html` and `http://machine.example.com/filename.html` from a local server, both URLs might appear in the cache. Because these files are from a local server, they may be retrieved so quickly that it is not necessary to cache them anyway.

However, if your company has servers in many remote locations, you may want to cache documents from all hosts to reduce network traffic and decrease the time needed to access the files.

To enable the caching of local hosts

1. Access the Server Manager, and click the Caching tab.
2. Click the Cache Local Hosts link. The Cache Local Hosts page displays.

3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression, and click OK.For more information on regular expressions, see .Chapter 16, "Managing Templates and Resources" on page 349.

4. Click the enabled button.

5. Click OK.

6. Click Restart Required. The Apply Changes page displays.

7. Click the Restart Proxy Server button to apply the changes.

Configuring the File Cache

The file cache is turned on by default. The file cache settings are contained in the `server.xml` file. You can use the Server Manager to change the file cache settings.

NOTE The Configure File Cache page displays in the user interface but is not implemented in this release of Proxy Server 4

To configure the file cache

1. From the Server Manager, click the Preferences tab.

2. Click the File Cache Configuration link. The File Cache Configuration page displays.

3. Select Enable File Cache, if not already selected.

4. Choose whether to transmit files.

   When you enable Transmit File, the server caches open file descriptors 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 Transmit File is enabled, 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, Transmit File is enabled on Windows, and disabled on UNIX. On UNIX, only enable Transmit File for platforms that have native OS support for PR_TransmitFile, which currently includes HP-UX and AIX. It is not recommended for other UNIX/Linux platforms.

5. Enter a size for the hash table.The default size is twice the maximum number of files plus 1. For example, if your maximum number of files is set to 1024, the default hash table size is 2049.

6. Enter a maximum age in seconds for a valid cache entry. By default, this is set to 30. 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 the maximum age based on whether the content is updated (existing files are modified) on a regular schedule. For example, if content is updated four times a day at regular intervals, you could set the maximum age to 21600 seconds (6 hours). Otherwise, consider setting the maximum age to the longest time you are willing to serve the previous version of a content file after the file has been modified.

7. Enter the Maximum Number of Files to be cached. By default, this is set to 1024.

8. Enter medium and small file size limits in bytes. By default, the Medium File Size Limit is set to 537600. By default, the Small File Size Limit is set to 2048.

   The cache treats small, medium, and large files differently. The contents of medium files are cached by mapping the file into virtual memory (currently only on UNIX/Linux platforms). The contents of small files are cached by allocating heap space and reading the file into it. The contents of large files (larger than medium) are not cached, although information about large files is cached. The advantage 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 Small File Size Limit is typically a slightly lower value than the VM page size.

9. Set the medium and small file space. The medium file space is the size in bytes of the virtual memory used to map all medium sized files. By default, this is set to 10485760. The small file space is 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 1048576 for UNIX/Linux.

10. Click OK.

11. Click Restart Required. The Apply Changes page displays.

12. Click the Restart Proxy Server button to apply the changes.

Viewing the URL Database

You can view the names and attributes of all recorded cached URLs. The URL information displayed is a list of cached documents grouped by access protocol and site name. You can limit the URLs you view in the list by typing a domain name into the Search field. By accessing this information, you can perform various cache management functions such as expiring and removing documents from the cache.

To view the URLs in the database

1. Access the Server Manager, and click the Caching tab.
2. Click the View URL Database link. The View URL Database page displays.
3. Click the Regenerate button to generate a current list of cached URLs. If you would like to view the information for a specific URL, enter a URL or regular expression in the Search field and click the Search button.
4. If you would like to view cache database information grouped by domain name and host, select a domain name from the list. A list of hosts in that domain appears. Click the name of a host and a list of URLs appears.
5. Click on the name of a URL. Detailed information about that URL appears.

Expiring and Removing Files from the Cache

From the View URL Database page you can expire and remove documents from the cache.

To expire or remove cached URLs

1. Access the Server Manager, and click the Caching tab.
2. Click the View URL Database link. The View URL Database page displays.
3. Click the Regenerate button. It generates a snapshot of the cache database. This snapshot forms the basis for the remaining steps.
4. If you know of a specific URL that you would like to expire or remove, enter that URL or a regular expression that matches that URL in the Search field and click the Search button. If you would like to work with URLs grouped by domain name and host, select a domain name from the list. A list of hosts in that domain appears. Click on the name of a host and a list of URLs appears.

5. To expire individual files, select the Ex option next to the URLs for those files and click the Exp / Rem Marked button. To expire all of the files in the list, click the Exp All button on the bottom of the form. To remove individual files from the cache, select the Rm option next to the URLs for those files and click the Exp / Rem Marked button. To remove all of the files in the list, click the Rem All button.

6. Click the Regenerate button to regenerate the snapshot.

NOTE When you use the Ex or Rm option, the associated file is processed but the changes are not reflected in the snapshot. The snapshot needs to be regenerated for the changes to be visible.

Using Cache Batch Updates

The Cache Batch Update feature allows you to pre-load files in a specified web site or do an up-to-date check on documents already in the cache whenever the proxy server is not busy. From the Set Cache Batch Updates page, you can create, edit, and delete batches of URLs and enable and disable batch updating.

Creating Batch Updates

You can actively (as opposed to on-demand) cache files by specifying files to be batch updated. The proxy server allows you to perform an up-to-date check on several files currently in the cache or pre-load multiple files in a particular web site.

To create a batch update

1. Access the Server Manager, and click the Caching tab.

2. Click the Set Cache Batch Updates link. The Set Cache Batch Updates page displays.

3. Select New and Create from the drop-down lists next to Create / Select a Batch Update Configuration.

4. Click OK. The Set Cache Batch Updates page displays.

5. In the Name section, enter a name for the new batch update entry.

6. In the Source section of the page, click the radio button for the type of batch update that you want to create. Click the first radio button if you want to perform an up-to-date check on all documents in the cache. Click the second radio button if you want to cache URLs recursively starting from the given source URL.
7. In the Source section fields, identify the documents that you want to use in the batch update.
8. In the Exceptions section, identify any files that you would like to exclude from the batch update.
9. In the Resources section, enter the maximum number of simultaneous connections and the maximum number of documents to traverse.
10. In the Timing section, enter the start and end times for the generation of the batch update. Only one batch update can be active at any time, so it is best to not overlap other batch update configurations.
11. Click OK.

NOTE You can create, edit, and delete batch update configurations without having batch updates turned on. However, if you want your batch updates to be updated according to the times you set on the Set Cache Batch Updates page, you must turn updates on.

12. Click Restart Required. The Apply Changes page displays.
13. Click the Restart Proxy Server button to apply the changes.

Editing or Deleting Batch Update Configurations

You can edit or delete batch updates using the Set Cache Batch Updates page. You may want to edit a batch update if you need to exclude certain files or want to update the batch more frequently. You may also want to delete a batch update configuration completely.

To edit or delete a batch update configuration

1. Access the Server Manager, and click the Caching tab.
2. Click the Set Cache Batch Updates link. The Set Cache Batch Updates page displays.

3. If you want to edit a batch, select the name of that batch and “Edit” from the drop-down lists next to Create / Select a Batch Update Configuration. If you want to delete a batch, select the name of that batch and “Delete” from the drop-down lists.
4. Click OK. The Set Cache Batch Updates page displays.
5. Modify the information as you wish.
6. Click OK.
7. Click Restart Required. The Apply Changes page displays.
8. Click the Restart Proxy Server button to apply the changes.

Using the Cache Command Line Interface

The proxy server comes with several command line utilities that let you configure, change, generate, and repair your cache directory structure. Most of these utilities are duplications of the Server Manager pages, but you might want to use the utilities if you need to schedule the maintenance (for example, as a cron job). All of the utilities are located in the `extras` directory.

To run the command line utilities

1. From the command-line prompt, go to the *server_root*/`proxy-server`*id* directory.
2. Type **`./start -shell`**

   The following sections describe the various utilities.

Building the Cache Directory Structure

The proxy has a utility called `cbuild` which is an offline cache database manager. This utility allows you create a new cache structure or modify an existing cache structure using the command-line interface. You can use the Server Manager pages to enable the proxy to use the newly created cache.The utility does not update the `server.xml` file. `cbuild` cannot resize a cache that has multiple partitions. The `server.xml` file has a element called `CACHE` that has a `cachecapacity` parameter. When the cache is created or modified by `cbuild`, the `cachecapacity` parameter should be manually updated in the `server.xml` file.

```
<PARTITION partitionname="part1" partitiondir="/home/build/install9
```

```
/proxy-server1/cache" maxsize="1600" minspace="5" enabled="true"/>
<CACHE enabled="true" cachecapacity="2000" cachedir="/tmp/cache">
```

You can invoke the `cbuild` utility in two modes. The first mode is:

`cbuild -d` *conf-dir* `-c` *cache-dir* `-s` *cache size*

`cbuild -d` *conf-dir* `-c` *cache-dir* `-s` *cache size* `-r`

For example:

`cbuild -d` *server_root*`/proxy-server`*id*`/config -c` *server_root*`/proxy-server`*id*`/cache -s 512`

`cbuild -d` *server_root*`/proxy-server`*id*`/config -c` *server_root*`/proxy-server`*id*`/cache -s 512 -r`

where:

- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*`/proxy-server`*id*`/config`.
- *cache-dir* is the directory for your cache structure.
- *cache size* is the maximum size to which the cache can grow This option cannot be used along with the *cache-dim* parameter. The maximum size is 65135 MB.
- `-r` resizes an existing cache structure provided it has a single partition. This is not required for creating a new cache.

The second mode you can run `cbuild` is:

`cbuild -d` *conf-dir* `-c` *cache-dir* `-n` *cache-dim*

`cbuild -d` *conf-dir* `-c` *cache-dir* `-n` *cache-dim* `-r`

For example:

`cbuild -d` *server_root*`/proxy-server`*id*`/config -c` *server_root*`/proxy-server`*id*`/cache -n 3`

`cbuild -d` *server_root*`/proxy-server`*id*`/config -c` *server_root*`/proxy-server`*id*`/cache -n 3 -r`

where:

- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*`/proxy-server`*id*`/config`.
- *cache-dir* is the directory for your cache structure.

- *cache-dim* determines the the number of sections. For example, in Figure 12-1 the section shown as s3.4, the 3 indicates the the dimension. The default value of *cache-dim* is 0 and the maximum value is 8.
- `-r` resizes an existing cache structure provided it has a single partition. This is not required for creating a new cache.

Managing the Cache URL List

The proxy has a utility called `urldb` that manages the URL list in the cache. You can use this utility to list the URLs that are cached. You can also selectively expire and remove cached objects from the cache database.

The `urldb` commands can be categorised into three groups based on the -o option:

- domains
- sites
- urls

To list domains, enter the following at the command line:

```
urldb -o matching_domains -e reg_exp -d conf-dir
```

For example:

```
urldb -o matching_domains -e “.*phoenix.*” -d
server_root/proxy-serverid/config
```

where

- `matching_domains` lists domains that match regular expression
- *reg_exp* is the regular expression used
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`.

To list all the matching sites in a domain, enter the following at the command line:

```
urldb -o matching_sites_in_domain -e reg_exp -m domain_name -d conf-dir
```

For example:

```
urldb -o matching_sites_in_domain -e “.*atlas” -m phoenix.com -d
server_root/proxy-serverid/config
```

where

- `matching_sites_in_domain` lists all the sites in a domain that match the regular expression
- *reg_exp* is the regular expression used
- *domain_name* is the name of the domain
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`

To list all the matching sites, enter the following at the command line:

```
urldb -o all_matching_sites -e reg_exp -d conf-dir
```

For example:

```
urldb -o all_matching_sites -e “.*atlas.*” -d server_root/proxy-serverid/config
```

where

- `all_matching_sites` lists all the sites that match the regular expression
- *reg_exp* is the regular expression used
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`

To list matching urls in a site, enter the following at the command line:

```
urldb -o matching_urls_from_site -e reg_exp -s site_name -d conf-dir
```

For example:

```
urldb -o matching_urls_from_site -e “http://.*atlas.*” -s atlas.phoenix.com
-d server_root/proxy-serverid/config
```

where

- `matching_urls_from_site` lists all urls from site that match the regular expression
- *reg_exp* is the regular expression used
- *site_name* is the name of the site
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`

To expire or remove matching urls in a site, enter the following at the command line:

```
urldb -o matching_urls_from_site -e reg_exp -s site_name -x e -d conf-dir
urldb -o matching_urls_from_site -e reg_exp -s site_name -x r -d conf-dir
```

For example:

```
urldb -o matching_urls_from_site -e “http://.*atlas.*” -s atlas.phoenix.com
-x e -d iserver_root/proxy-serverid/config
```

where

- `matching_urls_from_site` lists all urls from site that match the regular expression
- *reg_exp* is the regular expression used
- *site_name* is the name of the site
- `-x e` is the option to expire the matching URLs from the cache database. This option can not be used with the domain and site modes
- `-x r` is the option to remove the matching URLs from the cache database
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`

To list all matching urls , enter the following at the command line:

```
urldb -o all_matching_urls -e reg_exp -d conf-dir
```

For example:

```
urldb -o all_matching_urls -e “.*cgi-bin.*” -d
server_root/proxy-serverid/config
```

where

- `all_matching_urls` lists all the URLs that match the regular expression
- *reg_exp* is the regular expression used
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`

To expire or remove all matching urls , enter the following at the command line:

```
urldb -o all_matching_urls -e reg_exp -x e -d conf-dir
```

```
urldb -o all_matching_urls -e reg_exp -x r -d conf-dir
```

For example:

```
urldb -o all_matching_urls -e “.*cgi-bin.*” -x e -d
server_root/proxy-serverid/config
```

where

- `all_matching_urls` lists all the URLs that match the regular expression
- *reg_exp* is the regular expression used
- `-x e` is the option to expire the matching URLs from the cache database
- `-x r` is the option to remove the matching URLs from the cache database
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`

To expire or remove a list of URLs , enter the following at the command line:

`urldb -l` *url-list* `-x e -e` *reg_exp* `-d` *conf-dir*

`urldb -l` *url-list* `-x r -e` *reg_exp* `-d` *conf-dir*

For example:

`urldb -l url.lst -x e -e ".*cgi-bin.*" -d` *server_root*/`proxy-server`*id*/`config`

where

- `url-list` is the list of URLs that need to be expired. This option can be used for providing the URL list.
- `-x e` is the option to expire the matching URLs from the cache database.
- `-x r` is the option to remove the matching URLs from the cache database.
- *reg_exp* is the regular expression used
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`.

Managing Cache Garbage Collection

The `cachegc` utility allows you to clean up the cache database of objects may have expired or are too old to be cached in the directory due to cache size constraints.

NOTE Ensure that the CacheGC is not running in the proxy instance when the `cachegc` utility is used.

The `cachegc` utility can be used in the following way:

`cachegc -f` *leave-fs-full-percent* `-u` *gc-high-margin-percent* `-l` *gc-low-margin-percen*`t -e` *extra-margin-percen*`t -d` *conf-dir*

For example:

```
cachegc -f 50 -u 80 -l 60 -e 5 -d server_root/proxy-serverid/config
```

where

- *leave-fs-full-percent* determines the percentage of the cache partition size below which garbage collection will not go
- *gc-high-margin-percent* controls the percentage of the maximum cache size that, when reached, triggers garbage collection
- *gc-low-margin-percent* controls the percentage of the maximum cache size that the garbage collector targets
- *extra-margin-percent* is used by the garbage collector to determine the fraction of the cache to remove.
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`.

Managing Batch Updates

The `bu` utility updates the cache and works in two modes. In the first mode, it iterates through the cache database and updates all the URLs that are present in the cache by sending HTTP requests for each. In the second mode, it starts with a given URL and does a breadth first iteration of all the links from that URL to the depth that you specify and fetches pages to the cache. `bu` is a RFC compliant robot.

```
bu -n hostname -p port -t time-lmt -f contact-address -s sleep-time -o object -r n -d conf-dir
```

For example:

```
bu -n phoenix -p 80 -t 3600 -f admin@phoenix.com -s 60 -o nova -r n -d
server_root/proxy-serverid/config
```

where

- *hostname* is the hostname of the machine on which proxy is running. The default value is the localhost.
- *port* is the port on which proxy server is running. The default port is `8080`.
- *time-lmt* is the time limit to which the utility will run
- *contact-address* determines the contact address that would be sent in the HTTP requests that are sent from `bu`. The default value is `worm@proxy-name`.
- *sleep-time* is the sleep time between two consecutive requests. The default value is 5 seconds.

- *object* is the object specified in `bu.conf` that we are executing currently
- `-r n` option determines whether the `robot.txt` policy is followed. The default value is `y`.
- *conf-dir* is the configuration directory of the proxy instance. It is located in the following path *server_root*/`proxy-server`*id*/`config`.

Using the Internet Cache Protocol (ICP)

About ICP

The Internet Cache Protocol (ICP) is an object location protocol that enables caches to communicate with one another. Caches can use ICP to send queries and replies about the existence of cached URLs and about the best locations from which to retrieve those URLs. In a typical ICP exchange, one cache will send an ICP query about a particular URL to all neighboring caches. Those caches will then send back ICP replies that indicate whether or not they contain that URL. If they do not contain the URL, they send back a “MISS.” If they do contain the URL, they send back a “HIT.”

Routing through ICP Neighborhoods

ICP can be used for communication among proxies located in different administrative domains. It allows a proxy cache in one administrative domain to communicate with a proxy cache in another administrative domain. It is effective for situations in which several proxy servers want to communicate, but cannot all be configured from one master proxy (as they are in a proxy array). Figure 12-3 shows an ICP exchange between proxies in different administrative domains.

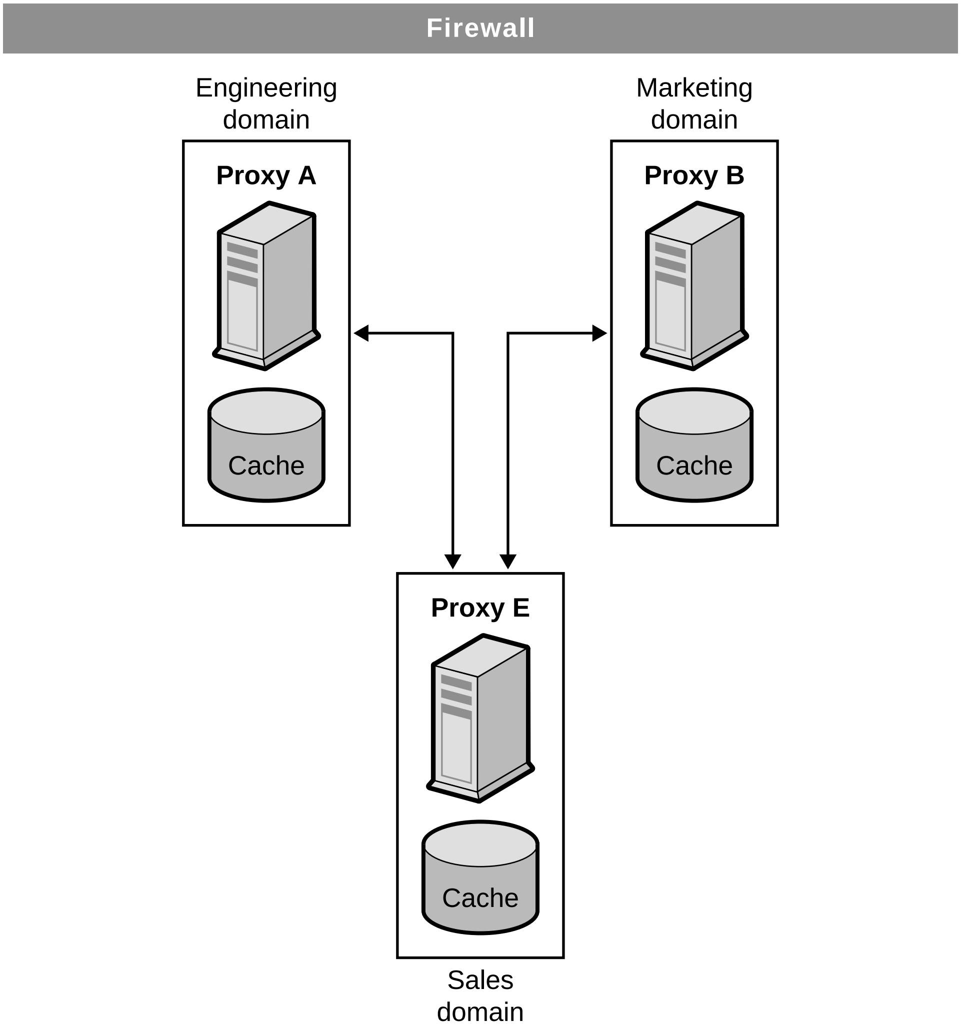
The proxies that communicate with each other through ICP are called *neighbors*. You cannot have more than 64 neighbors in an ICP neighborhood. There are two types of neighbors in an ICP neighborhood, *parents* and *siblings*. Only parents can access the remote server if no other neighbors have the requested URL. Your ICP neighborhood can have no parents or it can have more than one parent. Any neighbor in an ICP neighborhood that is *not* a parent is considered a sibling. Siblings cannot retrieve documents from remote servers unless the sibling is marked as the default route for ICP, and ICP uses the default.

You can use *polling rounds* to determine the order in which neighbors receive queries. A polling round is an ICP query cycle. For each neighbor, you must assign a polling round. If you configure all neighbors to be in polling round one, then all neighbors will be queried in one cycle. In other words, they will all be queried at the same time. If you configure some of the neighbors to be in polling round 2, then all of the neighbors in polling round one will be queried first and if none of them return a “HIT,” all round two proxies will be queried. The maximum number of polling rounds is two.

Since ICP parents are likely to be network bottlenecks, you can use polling rounds to lighten their load. A common setup is to configure all siblings to be in polling round one and all parents to be in polling round two. That way, when the local proxy requests a URL, the request goes to all of the siblings in the neighborhood first. If none of the siblings have the requested URL, the request goes to the parent. If the parent does not have the URL, it will retrieve it from a remote server.

Each neighbor in an ICP neighborhood must have at least one ICP server running. If a neighbor does not have an ICP server running, it cannot answer the ICP requests from their neighbors. Enabling ICP on your proxy server starts the ICP server if it is not already running.

Figure 12-3 An ICP Exchange



To set up ICP

1. Add parent(s) to your ICP neighborhood. (This step is only necessary if you want parents in your ICP neighborhood.) For more information on adding parents to an ICP neighborhood, see "Adding Parents to an ICP Neighborhood" on page 282.

2. Add sibling(s) to your ICP neighborhood. For more information on adding siblings to your ICP neighborhood, see "Adding Siblings to an ICP Neighborhood" on page 284.

3. Configure each neighbor in the ICP neighborhood. For more information on configuring ICP neighbors, see "Configuring Individual ICP Neighbors" on page 286.

4. Enable ICP. For information on enabling ICP, see "Enabling ICP" on page 287.

5. If your proxy has siblings or parents in its ICP neighborhood, enable routing through an ICP neighborhood. For more information on enabling routing through an ICP neighborhood, see "Enabling Routing Through an ICP Neighborhood" on page 288.

Adding Parents to an ICP Neighborhood

To add parent proxies to an ICP neighborhood

1. Access the Server Manager, and click the Caching tab.

2. Click the Configure ICP link. The Configure ICP page displays.

3. In the Parent List section of the page, click the Add button. The ICP Parent page displays.

4. In the Machine Address field, enter the IP address or host name of the parent proxy you are adding to the ICP neighborhood.

5. In the ICP Port field, enter the port number on which the parent proxy will listen for ICP messages.

6. In the Multicast Address field, you can enter the multicast address to which the parent listens. A multicast address is an IP address to which multiple servers can listen. Using a multicast address allows a proxy to send one query to the network that all neighbors who are listening to that multicast address can see; therefore, eliminating the need to send a query to each neighbor separately. Using multicast is optional.

NOTE Neighbors in different polling rounds should not listen to the same multicast address.

7. In the TTL field, enter the number of subnets that the multicast message will be forwarded to. If the TTL is set to 1, the multicast message will only be forwarded to the local subnet. If the TTL is 2, the message will go to all subnets that are one level away, and so on.

NOTE Multicast makes it possible for two unrelated neighbors to send ICP messages to eachother. Therefore, if you want to prevent unrelated neighbors from receiving ICP messages from the proxies in your ICP neighborhood, you should set a low TTL value in the TTL field.

8. In the Proxy Port field, enter the port for the proxy server on the parent.
9. From the Polling Round drop-down list, choose the polling round that you want the parent to be in. The default polling round is 1.
10. Click OK.
11. Click Restart Required. The Apply Changes page displays.
12. Click the Restart Proxy Server button to apply the changes.

Editing Parent Configurations in an ICP Neighborhood

To edit the parent configuration

1. Access the Server Manager, and click the Caching tab.
2. Select the Configure ICP link. The Configure ICP page displays.
3. Click the radio button next to the parent you want to edit.
4. Click the Edit button.
5. Modify the appropriate information.
6. Click OK.
7. Click Restart Required. The Apply Changes page displays.
8. Click the Restart Proxy Server button to apply the changes.

Removing Parents from an ICP Neighborhood

To remove parent proxies from an ICP neighborhood

1. Access the Server Manager, and click the Caching tab.
2. Select the Configure ICP link. The Configure ICP page displays.
3. Click the radio button next to the parent you want to remove.
4. Click the Delete button.
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Adding Siblings to an ICP Neighborhood

To add sibling proxies to an ICP neighborhood

1. Access the Server Manager, and click the Caching tab.
2. Select the Configure ICP link. The Configure ICP page displays.
3. In the Sibling List section of the page, click the Add button. The ICP Sibling page displays.
4. In the Machine Address field, enter the IP address or host name of the sibling proxy you are adding to the ICP neighborhood.
5. In the Port field, enter the port number on which the sibling proxy will listen for ICP messages.
6. In the Multicast Address field, enter the multicast address to which the sibling listens. A multicast address is an IP address to which multiple servers can listen. Using a multicast address allows a proxy to send one query to the network that all neighbors who are listening to that multicast address can see; therefore, eliminating the need to send a query to each neighbor separately.

NOTE Neighbors in different polling rounds should not listen to the same multicast address.

7. In the TTL field, enter the number of subnets that the multicast message will be forwarded to. If the TTL is set to 1, the multicast message will only be forwarded to the local subnet. If the TTL is 2, the message will go to all subnets that are one level away.

NOTE Multicast makes it possible for two unrelated neighbors to send ICP messages to eachother. Therefore, if you want to prevent unrelated neighbors from receiving ICP messages from the proxies in your ICP neighborhood, you should set a low TTL value in the TTL field.

8. In the Proxy Port field, enter the port for the proxy server on the sibling.
9. From the Polling Round down-down list, choose the polling round that you want the sibling to be in. The default polling round is 1.
10. Click OK.
11. Click Restart Required. The Apply Changes page displays.
12. Click the Restart Proxy Server button to apply the changes.

Editing Sibling Configurations in an ICP Neighborhood

To edit the sibling configuration

1. Access the Server Manager, and click the Caching tab.
2. Select the Configure ICP link. The Configure ICP page displays.
3. Click the radio button next to the sibling you want to edit.
4. Click the Edit button.
5. Modify the appropriate information.
6. Click OK.
7. Click Restart Required. The Apply Changes page displays.
8. Click the Restart Proxy Server button to apply the changes.

Removing Siblings from an ICP Neighborhood

To remove sibling proxies from an ICP neighborhood

1. Access the Server Manager, and click the Caching tab.
2. Select the Configure ICP link. The Configure ICP page displays.
3. Click the radio button next to the sibling you want to remove.
4. Click the Delete button.
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Configuring Individual ICP Neighbors

You need to configure each neighbor, or local proxy, in your ICP neighborhood.

To configure the local proxy server in your ICP neighborhood

1. Access the Server Manager, and click the Caching tab.
2. Select the Configure ICP link. The Configure ICP page displays.
3. In the Binding Address field, enter the IP address to which the neighbor server will bind.
4. In the Port field, enter the port number to which the neighbor server will listen for ICP.
5. In the Multicast Address field, enter the multicast address to which the neighbor listens. A multicast address is an IP address to which multiple servers can listen. Using a multicast address allows a proxy to send one query to the network that all neighbors who are listening to that multicast address can see; therefore, eliminating the need to send a query to each neighbor separately.

   If both a multicast address and bind address are specified for the neighbor, the neighbor uses the bind address to send replies and uses multicast to listen. If neither a bind address or a multicast address is specified, the operating system will decide which address to use to send the data.
6. In the Default Route field, enter the name or IP address of the proxy to which the neighbor should route a request when none of the neighboring proxies respond with a “hit.” If you enter the word “origin” into this field, or if you leave it blank, the default route will be to the origin server.

NOTE If you choose “first responding parent” from the No Hit Behavior drop-down list , the route you enter in the Default Route field will have no effect. The proxy only uses this route if you choose the default no hit behavior.

7. In the second Port field, enter the port number of the default route machine that you entered into the Default Route field.
8. From theOn No Hits, Route Through drop-down list, choose the neighbor’s behavior when none of the siblings in the ICP neighborhood have the requested URL in their caches. You can choose:
    - **first responding parent.** The neighbor will retrieve the requested URL through the parent that first responds with a “miss”
    - **default route**. The neighbor will retrieve the requested URL through the machine specified in the Default Route field.
9. In the Server Count field, enter the number of processes that will service ICP requests.
10. In the Timeout field, enter the maximum amount of time the neighbor will wait for an ICP response in each round.
11. Click OK.
12. Click Restart Required. The Apply Changes page displays.
13. Click the Restart Proxy Server button to apply the changes.

Enabling ICP

To enable ICP

1. Access the Server Manager, and click the Preferences tab.
2. Click the Configure System Preferences link. The Configure System Preferences page displays.
3. Select the Yes radio button for ICP.
4. Click OK.
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Enabling Routing Through an ICP Neighborhood

To enable routing through an ICP neighborhood

1. Access the Server Manager, and click the Routing tab.
2. Click the Set Routing Preferences link. The Set Routing Preferences page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, and enter a regular expression, and click OK.
4. Select the radio button next to the text Route Through.
5. Select the checkbox next to ICP.
6. If you want the client to retrieve a document directly from the ICP neighbor that has the document instead of going through another neighbor to get it, select the checkbox next to the text redirect.
7. Click OK.

CAUTION Redirect is not currently supported by any clients, so don’t use the feature at this time.

NOTE You need to enable routing through an ICP neighborhood only if your proxy has other siblings or parents in the ICP neighborhood. If your proxy is a parent to another proxy and does not have any siblings or parents of its own, then you need to enable ICP only for that proxy. You do not need to enable routing through an ICP neighborhood.

8. Click Restart Required. The Apply Changes page displays.
9. Click the Restart Proxy Server button to apply the changes.

Using Proxy Arrays

About Proxy Arrays

Proxy arrays for distributed caching allow multiple proxies to serve as a single cache. In other words, each proxy in the array will contain different cached URLs that can be retrieved by a browser or downstream proxy server. Proxy arrays prevent the duplication of caches that often occurs with multiple proxy servers. Through hash-based routing, proxy arrays route requests to the correct cache in the proxy array.

Proxy arrays also allow incremental scalability. In other words, if you decide to add another proxy to your proxy array, each member's cache is not invalidated. Only 1/*n* of the URLs in each member's cache, where *n* is the number of proxies in your array, will be reassigned to other members.

Routing through Proxy Arrays

For each request through a proxy array, a hash function assigns each proxy in the array a score that is based on the requested URL, the proxy's name and the proxy's load factor. The request is then routed to the proxy with the highest score.

Since requests for URLs can come from both clients and proxies, there are two types of routing through proxy arrays: *client to proxy routing* and *proxy to proxy routing*.

In client to proxy routing, the client uses the Proxy Auto Configuration (PAC) mechanism to determine which proxy to go through. However, instead of using the standard PAC file, the client uses a special PAC file which computes the hash algorithm to determine the appropriate route for the requested URL, Figure 12-4 shows client to proxy routing.

In Figure 12-4, each member of the proxy array loads and polls the master proxy for updates to the PAT file. Once the client has a PAC file, it only needs to download this file again if the configuration changes. Generally, clients will download the PAC file at restart.

The proxy server can automatically generate the special PAC file from the Proxy Array Membership Table (PAT) specifications made through the administration interface.

In proxy to proxy routing, proxies use a PAT (Proxy Array Table) file to compute the hash algorithm instead of the PAC file used by clients. The PAT file is an ASCII file that contains information about a proxy array, including the proxies' machine names, IP addresses, ports, load factors, cache sizes, etc. For computing the hash algorithm at the server, it is much more efficient to use a PAT file than a PAC file (which is a JavaScript file that has to be interpreted at run-time). However, most clients do not recognize the PAT file format, and therefore, must use a PAC file. Figure 12-5 shows proxy to proxy routing.

The PAT file will be created on one proxy in the proxy array - the master proxy. The proxy administrator must determine which proxy will be the master proxy. The administrator can change the PAT file from this master proxy server and all other members of the proxy array can then manually or automatically poll the master proxy for these changes. You can configure each member to automatically generate a PAC file from these changes.

You can also chain proxy arrays together for hierarchical routing. If a proxy server routes an incoming request through an upstream proxy array, the upstream proxy array is then known as a parent array. A parent array is a proxy array that a proxy server goes through. In other words, if a client requests a document from Proxy X, and Proxy X does not have the document, it sends the request to Proxy Array Y instead of sending it directly to the remote server. So, Proxy Array Y is a parent array. In Figure 12-5, Proxy Array 1 is a parent array to Proxy Array 2. A member of Proxy Array 2 loads and polls for updates to the parent array's PAT file. Usually, it polls the master proxy in the parent array. The hash algorithm for the requested URL is computed using the downloaded PAT file and the member in the Proxy Array 2 then retrieves the requested URL from whichever proxy in Proxy Array1 that has the highest score. In Figure 12-5, Proxy B has the highest score for the URL requested by the client.

Figure 12-4 Client to Proxy Routing

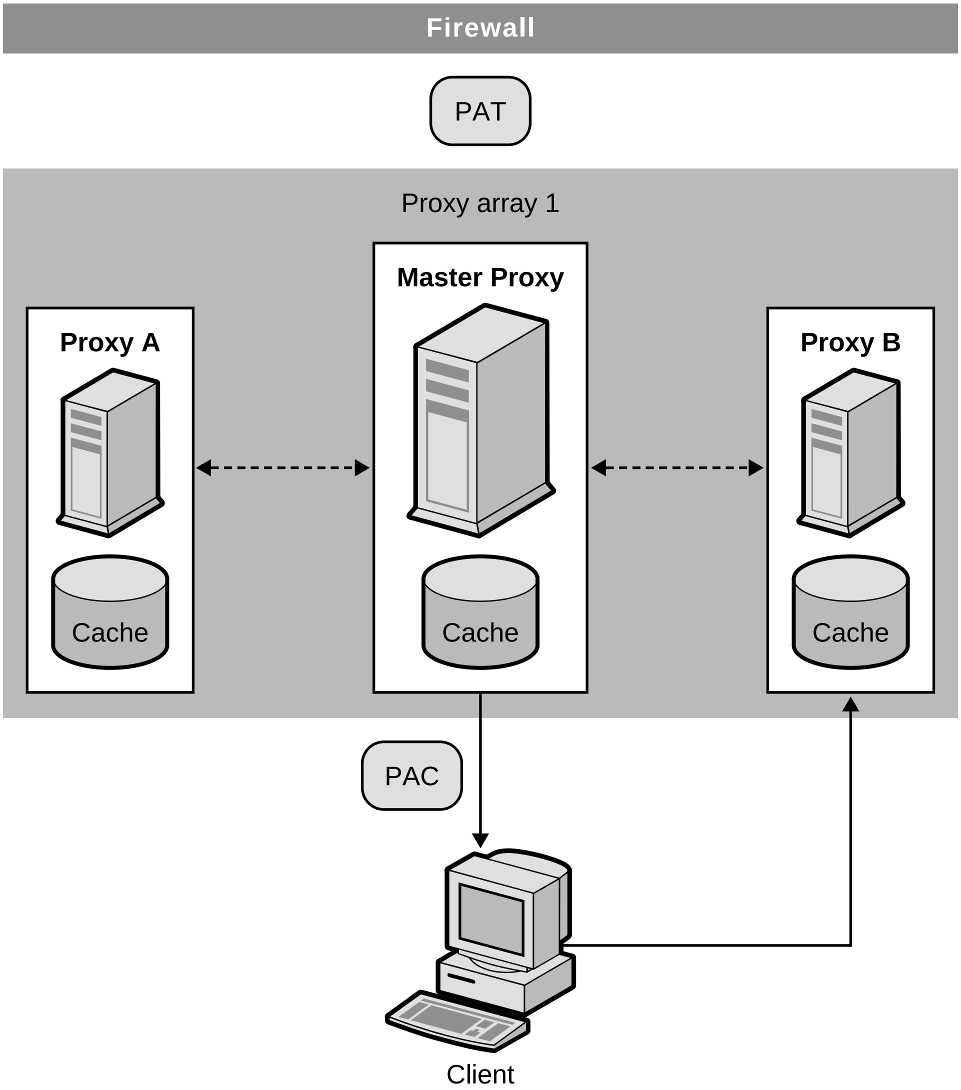
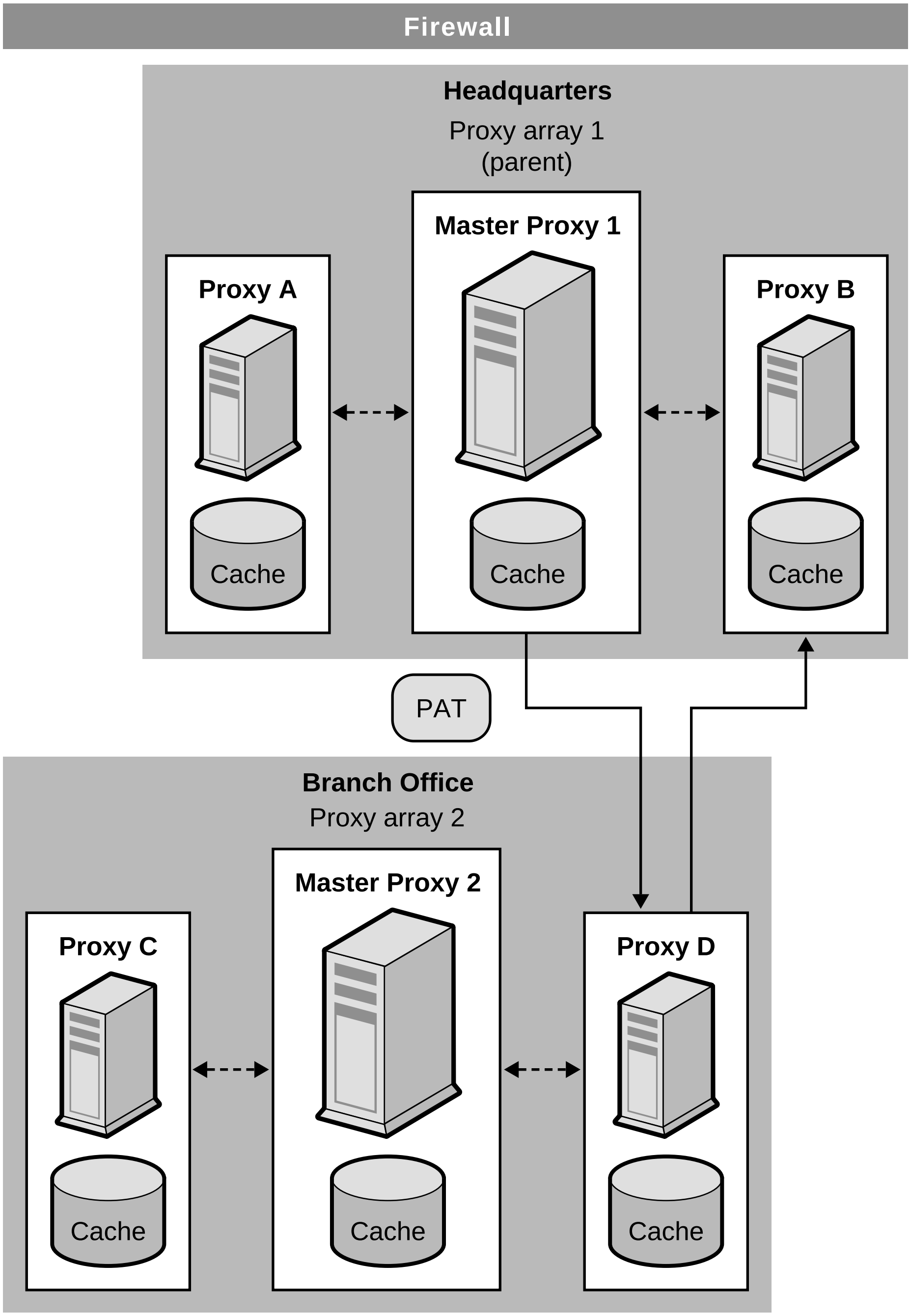


Figure 12-5 Proxy to Proxy Routing



To set up a proxy array

1. From the master proxy, do the following steps:
    - **a.** Create the proxy array. For more information on creating the member list, see "Creating a Proxy Array Member List" on page 294.
    - **b.** Generate a PAC file from your PAT file. You only need to generate a PAC file if you are using client to proxy routing. For more information on generating a PAC file from a PAT file, see "Generating a PAC File from a PAT File" on page 299.
    - **c.** Configure the master member of the array. For more information on configuring master members, see "Configuring Proxy Array Members" on page 296.
    - **d.** Enable routing through a proxy array. For more information on enabling routing through a proxy array, see "Enabling Routing Through a Proxy Array" on page 297.
    - **e.** Create a PAT mapping to map the URL "/pat" to the PAT file.
    - **f.** Enable your proxy array. For more information on enabling a proxy array, see "Enabling a Proxy Array" on page 298.
2. From each of the non-master proxies, do the following steps:
    - **a.** Configure the non-master member of the array. For more information on configuring non-master members, see "Configuring Proxy Array Members" on page 296
    - **b.** Enable routing through a proxy array. For more information on enabling routing through a proxy array, see "Enabling Routing Through a Proxy Array" on page 297.
    - **c.** Enable your proxy array. For more information on enabling a proxy array, see "Enabling a Proxy Array" on page 298.

.

NOTE If your proxy array is going to route through a parent array, you also need to enable the parent array and configure each member to route through a parent array for desired URLs. For more information on parent arrays, see "Routing through Parent Arrays" on page 301.

Creating a Proxy Array Member List

You should create and update the proxy array member list from the master proxy of the array only. You only need to create the proxy array member list once, but you can modify it at any time. By creating the proxy array member list, you are generating the PAT file to be distributed to all of the proxies in the array and to any downstream proxies.

CAUTION You should only make changes or additions to the proxy array member list through the master proxy in the array. All other members of the array can only read the member list.

1. Access the Server Manager, and click the Caching tab
2. Click the Configure Proxy Array link. The Configure Proxy Array page displays.
3. In the Array name field, enter the name of the array.
4. In the Reload Configuration Every field, enter the number of minutes between each polling for the PAT file.
5. Click the Array Enabled checkbox.
6. Click the Create button.

NOTE Be sure to click OK before you begin to add members to the member list.

NOTE The Create button changes to an OK button after the proxy array has been created.

7. Click Restart Required. The Apply Changes page displays.
8. For each member in the proxy array, enter the following and then click OK:
    - **Name.** The name of the proxy server you are adding to the member list
    - **IP Address.** The IP address of the proxy server you are adding to the member list
    - **Port.** This is the port on which the member polls for the PAT file.
    - **Load Factor.** An integer that reflects the relative load that should be routed through the member.

- **Status.** The status of the member. This value can be either on or off. If you disable a proxy array member, the member's requests will be re-routed through another member.

NOTE The master member should be added first before adding the other members.

NOTE Be sure to click OK after you enter the information for each proxy array member you are adding.

9. Click Restart Required. The Apply Changes page displays.
10. Click the Restart Proxy Server button to apply the changes.

Editing Proxy Array Member List Information

At any time, you can change the information for the members in the proxy array member list. You can only edit the proxy array member list from the master proxy.

CAUTION You should only make changes or additions to the proxy array member list through the master proxy in the array. If you modify this list from any other member of the array, all changes will be lost.

To edit member list information for any of the members in a proxy array

1. Access the Server Manager, and click the Caching tab.
2. Click the Configure Proxy Array link. The Configure Proxy Array page displays.
3. In the Member List, select the radio button next to the member that you want to edit.
4. Click the Edit button. The Configure Proxy Array Member page displays.
5. Edit the appropriate information.
6. Click OK.
7. Click Restart Required. The Apply Changes page displays.
8. Click the Restart Proxy Server button to apply the changes.

NOTE If you want your changes to take effect and to be distributed to the members of the proxy array, you need to update the Configuration ID on the Configure Proxy Array page and click OK. To update the configuration ID, you can simply increase it by one.

Deleting Proxy Array Members

Deleting proxy array members will remove them from the proxy array. You can only delete proxy array members from the master proxy.

CAUTION You should only make changes or additions to the proxy array member list through the master proxy in the array. If you modify this list from any other member of the array, all changes will be lost.

To delete members of a proxy array

1. Access the Server Manager, and click the Caching tab.
2. Click the Configure Proxy Array link. The Configure Proxy Array page displays.
3. In the Member List, select the radio button next to the member that you want to delete.
4. Click the Delete button.

NOTE If you want your changes to take effect and to be distributed to the members of the proxy array, you need to update the Configuration ID on the Configure Proxy Array page and click OK. To update the configuration ID, you can simply increase it by one.

5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Configuring Proxy Array Members

You only need to configure each member in the proxy array once, and you must do so from the member itself. You cannot configure a member of the array from another member. You also need to configure the master proxy.

To configure each member of the proxy array

1. Access the Server Manager, and click the Caching tab.
2. Click the Configure Proxy Array Member link. The Configure Proxy Array Member page displays.
3. In the Proxy Array section, indicate whether or not the member needs to poll for the PAT file by selecting the appropriate radio button. The choices are:
    - **Non-Master Member.** You should select this option if the member you are configuring is *not* the master proxy. Any proxy array member that is not a master proxy will need to poll for the PAT file in order to retrieve it from the master proxy.
    - **Master Member.** You should select this option if you are configuring the master proxy. If you are configuring the master proxy, the PAT file is local and does not need to be polled.
4. In the Poll Host field, enter the name of the master proxy that you will be polling for the PAT file.
5. In the Port field, enter the port at which the master proxy accepts HTTP requests.
6. In the URL field, enter the URL of the PAT file on the master proxy. If on your master proxy, you have created a PAT mapping to map the PAT file to the URL `/pat`, you should enter **`/pat`** in the URL field.
7. In the Headers File field, enter the full pathname for a file with any special headers that must be sent with the HTTP request for the PAT file (such as authentication information). This field is optional.
8. Click OK.
9. Click Restart Required. The Apply Changes page displays.
10. Click the Restart Proxy Server button to apply the changes.

Enabling Routing Through a Proxy Array

To enable routing through a proxy array

1. Access the Server Manager, and click the Routing tab.
2. Click the Set Routing Preferences link. The Set Routing Preferences page displays.

3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression, and click OK.
4. Select the Route Through option.
5. Select the checkboxes for proxy array and / or parent array.
6. If you choose to route through a proxy array and you want to redirect requests to another URL, select the redirect checkbox. Redirecting means that if a member of a proxy array receives a request that it should not service, it tells the client which proxy to contact for that request.
7. Click OK.

NOTE You can only enable proxy array routing if the proxy server you are configuring is a member of a proxy array. You can only enable parent routing if a parent array exists. Both routing options are independent of each other.

CAUTION Redirect is not currently supported by any clients, so you should not use the feature at this time.

8. Click Restart Required. The Apply Changes page displays.
9. Click the Restart Proxy Server button to apply the changes.

Enabling a Proxy Array

To enable a proxy array

1. Access the Server Manager, and clcik the Preferences tab.
2. Click the Configure System Preferences link. The Configure System Preferences page displays.
3. Click the Yes option for the type of array or arrays you want to enable - either a normal proxy array or a parent array.

4. Click OK.

NOTE If you are not routing through a proxy array, you should make sure that all clients use a special PAC file to route correctly before you disable the proxy array option. If you disable the parent array option, you should have valid alternative routing options set in the Set Routing Preferences page, such as explicit proxy or a direct connection.

5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Redirecting Requests in a Proxy Array

If you choose to route through a proxy array, you need to designate whether you want to redirect requests to another URL. Redirecting means that if a member of a proxy array receives a request that it should not service, it tells the client which proxy to contact for that request.

CAUTION Redirect is not currently supported by any clients, so you should not use the feature at this time.

Generating a PAC File from a PAT File

Because most clients do not recognize the PAT file format, the clients in client to proxy routing use the Proxy Auto Configuration (PAC) mechanism to receive information about which proxy to go through. However, instead of using the standard PAC file, the client uses a special PAC file derived from the PAT file. This special PAC file computes the hash algorithm to determine the appropriate route for the requested URL.

You can manually or automatically generate a PAC file from the PAT file. If you manually generate the PAC file from a specific member of the proxy array, that member will immediately re-generate the PAC file based on the information currently in the PAT file. If you configure a proxy array member to automatically generate a PAC file, the member will automatically re-generate the file after each time it detects a modified version of the PAT file.

NOTE If you are not using the proxy array feature for your proxy server, then you should use the Create / Edit Autoconfiguration File page to generate your PAC file. For more information see Chapter 17, "Using the Client Autoconfiguration File" on page 355.

Manually Generating a PAC File from a PAT File

NOTE The PAC file can be generated only from the master proxy.

To manually generate a PAC file from a PAT file

1. Access the Server Manager of the master proxy, and click the Caching tab.

2. Click the Configure Proxy Array link. The Configure Proxy Array page displays.

3. Click the Generate PAC button. The PAC Generation page displays.

4. If you want to use custom logic in your PAC file, in the Custom logic file field, enter the name of the file containing the customized logic you would like to include in the generation of your PAC file. This logic is inserted before the proxy array selection logic in the `FindProxyForURL` function. This function is typically used for local requests which need not go through the proxy array.

   If you have already entered the custom logic file on the Configure Proxy Array Member page, this field will be populated with that information. You may edit the custom logic filename if you wish, and the changes you make will transfer to the Configure Proxy Array Member page as well.

5. In the Default Route field, enter the route a client should take if the proxies in the array are not available.

   If you have already entered the default route on the Configure Proxy Array Member page, this field will be populated with that information. You may edit the default route if you wish, and the changes you make will transfer to the Configure Proxy Array Member page as well.

6. Click OK.

7. Click Restart Required. The Apply Changes page displays.
8. Click the Restart Proxy Server button to apply the changes.

Automatically Generating a PAC File from a PAT File

To automatically generate a PAC file from a PAT file each time a change is detected

1. Access the Server Manager, and click the Caching tab.
2. Click the Configure Proxy Array Member link.The Configure Proxy Array Member page displays.
3. Select the Auto-generate PAC File checkbox.
4. If you want to use custom logic in your PAC file, in the Custom Logic File field, enter the name of the file containing the customized logic you would like to include in the generation of your PAC file. This logic is inserted before the proxy array selection logic in the `FindProxyFor URL` function.

   If you have already entered and saved the custom logic file on the Configure Proxy Array page, this field will be populated with that information. You may edit the custom logic file name if you wish, and the changes you make will transfer to the Configure Proxy Array page as well.
5. In the Default Route field, enter the route a client should take if the proxies in the array are not available.
6. If you have already entered and saved the default route on the Configure Proxy Array page, this field will be populated with that information. You may edit the default route if you wish, and the changes you make will transfer to the Configure Proxy Array page as well.
7. Click OK.
8. Click Restart Required. The Apply Changes page displays.
9. Click the Restart Proxy Server button to apply the changes.

Routing through Parent Arrays

You can configure your proxy or proxy array member to route through an upstream parent array instead of going directly to a remote server.

To configure a proxy or proxy array member to route through a parent array

1. Enable the parent array. For more information on enabling an array, see "Enabling a Proxy Array" on page 298.

2. Enable routing through the parent array. For more information on enabling routing through an array, see “Enabling Routing Through a Proxy Array” on page 297.

3. Access the Server Manager, and click the Caching tab.

4. Click the Configure Proxy Array Member link. The Configure Proxy Array Member page displays.

5. In the Poll Host field in the Parent Array section of the page, enter the host name of the proxy in the parent array that you will poll for the PAT file. This proxy is usually the master proxy of the parent array.

6. In the Port field in the Parent Array section of the page, enter the Port number of the proxy in the parent array that you will poll for the PAT file.

7. In the URL field, enter the URL of the PAT file on the master proxy. If on your master proxy, you have created a PAT mapping, you should enter the mapping into this URL field.

8. In the Headers File field in the Parent Array section of the form, full pathname for a file with any special headers that must be sent with the HTTP request for the PAT file (such as authentication information). This field is optional.

9. Click OK.

10. Click Restart Required. The Apply Changes page displays.

11. Click the Restart Proxy Server button to apply the changes.

Viewing Parent Array Information

If your proxy array is routing through a parent array, you need information about the members of the parent array. This information is sent from the parent array in the form of a PAT file. The information in this PAT file is displayed on the View Parent Array Configuration page.

To view parent array information

1. Access the Server Manager, and click the Caching tab.

2. Click the View Parent Array Configuration link. The View Parent Array Configuration page displays.

3. View the information.

Chapter 13

Filtering Content through the Proxy

This chapter describes how to filter URLs so that your proxy server either does not allow access to the URL or modifies the HTML and JavaScript content it returns to the client. This chapter also describes how you can restrict access through the proxy based on the web browser (user agent) that the client is using.

The proxy server lets you use a URL filter file to determine which URLs the server supports. For example, instead of manually typing in wildcard patterns of URLs to support, you can create or purchase one text file that contains URLs you want to restrict. This feature lets you create one file of URLs that you can use on many different proxy servers.

You can also filter URLs based on their MIME type. For example, you might allow the proxy to cache and send HTML and GIF files but not allow it to get binary or executable files because of the risk of computer viruses.

This chapter contains the following sections:

- Filtering URLs
- Content URL Rewriting
- Restricting Access to Specific Web Browsers
- Blocking Requests
- Suppressing Outgoing Headers
- Filtering by MIME Type
- Filtering by HTML Tags
- Configuring the Server for Content Compression

Filtering URLs

You can use a file of URLs to configure what content the proxy server retrieves. You can set up a list of URLs the proxy always supports and a list of URLs the proxy never supports.

For example, if you are an Internet service provider who runs a proxy server with content appropriate for children, you might set up a list of URLs that are approved for viewing by children. You can then have the proxy server retrieve only the approved URLs; if a client tries to go to an unsupported URL, either you can have the proxy return the default “Forbidden” message or you can create a custom message explaining why the client could not access that URL.

To restrict access based on URLs, you need to create a file of URLs to allow or restrict. You can do this through the Server Manager. Once you have the file, you can set up the restrictions. These processes are discussed in the following sections.

Creating a Filter File of URLs

A filter file is a file that contains a list of URLs. The filter files the proxy server uses are plain text files with lines of URLs in the following pattern:

```
protocol://host:port/path/filename
```

You can use regular expressions in each of the three sections: `protocol`, `host:port`, and `path/filename`. For example, if you want to create a URL pattern for all protocols going to the netscape.com domain, you’d have the following line in your file:

```
.*://.*\.example\.com/.*
```

This line works only if you do not specify a port number. For more information on regular expressions, see “Understanding Regular Expressions” in Chapter 16, “Managing Templates and Resources” on page 349.

If you want to create your own file without using the Server Manager, you should use the Server Manager pages to create an empty file, and then add your text in that file or replace the file with one containing the regular expressions.

To create a filter file

1. Access the Server Manager, and click the Filters tab.
2. Click the Restrict URL Filter Access link. The Restrict URL Filter Access page displays.
3. Choose New Filter from the drop-down list next to the Create/Edit button.

4. Type a name for the filter file in the text box to the right of the drop-down list and then click the Create/Edit button. The Filter Editor page displays.
5. Use the Filter Content scrollable text box to enter URLs and regular expressions of URLs. The Reset button clears all the text in this field.

   For more information on regular expressions, see "Understanding Regular Expressions" in Chapter 16, "Managing Templates and Resources" on page 349.
6. Click OK.

The proxy server creates the file and returns you to the Restrict URL Filter Access page. The filter file is created in the `proxy-server`*id*`/conf_bk` directory.

Setting Default Access for a Filter File

Once you have a filter file that contains the URLs you want to use, you can set the default access for those URLs.

To set default access for a filter file

1. Access the Server Manager, and click the Filters tab.
2. Click the Restrict URL Filter Access link. The Restrict URL Filter Access page displays.
3. Choose the template you want to use with the filters.

   Typically, you will want to create filter files for the entire proxy server, but you might want one set of filter files for HTTP and another for FTP.
4. Use the URL Filter To Allow list to choose a filter file that contains the URLs you want the proxy server to support.
5. Use the URL Filter To Deny list to choose a filter file that contains the URLs to which you want the proxy server to deny access.
6. Choose the text you want the proxy server to return to clients who request a denied URL. You can choose one of two options:
   - You can send the default "Forbidden" response that the proxy generates.
   - You can send a text or HTML file with customized text. Type the absolute path to this file in the text box.
7. Click OK.
8. Click Restart Required. The Apply Changes page displays.

9. Click the Restart Proxy Server button to apply the changes.

Content URL Rewriting

Proxy Server 4 has the ability to inspect the content being returned to the client and replace patterns (such as URLs) with other strings. There are two parameters that can be configured - a source string and a destination string. The Proxy Server looks for text matching the source string and substitutes it with the text in the destination string. This feature works only in the reverse proxy mode.

To create a URL rewriting pattern

1. Access the Server Manager, and click the Filters tab.
2. Click the Set Content URL Rewriting link. The Set Content URL Rewriting page displays.
3. Select a resource from the drop-down list or specify a regular expression. For more information on regular expressions, see “Understanding Regular Expressions” in Chapter 16, “Managing Templates and Resources” on page 349.
4. Specify the source string in the Source Pattern text box.
5. Specify the destination string in the Destination Pattern text box.
6. Specify the content type in the MIME Pattern text box.
7. Click OK.
8. Click Restart Required. The Apply Changes page displays.
9. Click the Restart Proxy Server button to apply the changes.

To edit a URL rewriting pattern

1. Access the Server Manager, and click the Filters tab.
2. Click the Set Content URL Rewriting link. The Set Content URL Rewriting page displays.
3. Click the Edit link next to the URL rewriting pattern you want to edit.
4. Click OK
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

To delete a URL rewriting pattern

1. Access the Server Manager, and click the Filters tab.
2. Click the Set Content URL Rewriting link. The Set Content URL Rewriting page displays.
3. Click the Remove link next to the URL rewriting pattern you want to delete. Click OK to confirm deletion.
4. Click Restart Required. The Apply Changes page displays.
5. Click the Restart Proxy Server button to apply the changes.

Restricting Access to Specific Web Browsers

You can restrict access to the proxy server based on the type and version of the client's web browser. Restriction occurs based on the user-agent header that all web browsers send to servers when making requests.

To restrict access to the proxy based on the client's web browser

1. Access the Server Manager, and click the Filters tab.
2. Click the Set User-Agent Restriction link. The Set User-Agent Restriction page displays.
3. Select the resource from the drop-down list or type a regular expression that matches the user-agent string for the browsers you want the Proxy Server to support. If you want to specify more than one client, enclose the regular expression in parentheses and use the | character to separate the multiple entries. For more information on regular expressions, see "Understanding Regular Expressions" in Chapter 16, "Managing Templates and Resources" on page 349.
4. Check the Allow Only User-Agents Matching option.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Blocking Requests

You may want to block file uploads and other requests based on the upload content type.

To block requests based on MIME type

1. Access the Server Manager, and click the Filters tab.
2. Click the Set Request Blocking link. The Set Request Blocking page displays.
3. Select the resource from the drop-down list or click the Regular Expression button, enter a regular expression and click OK.
4. Click the radio button for the type of request blocking you want. The options include the following:
   - Disabled - disables request blocking
   - Multipart MIME (File Upload) - blocks all file uploads
   - MIME Types Matching Regular Expression - blocks requests for MIME types that match the regular expression you enter. For more information on regular expressions, see "Understanding Regular Expressions" in Chapter 16, "Managing Templates and Resources" on page 349.
5. Choose whether you want to block requests for all clients or for user-agents that match a regular expression you enter.
6. Click the radio button for the methods for which you want to block requests. The options are:
   - Any Method With Request Body - blocks all requests with a request body, regardless of the method
   - only for:
     POST - blocks file upload requests using the POST method
     PUT - blocks file upload requests using the PUT method
   - Methods Matching Regular Expression- blocks all file upload requests using the method you enter
7. Click OK.
8. Click Restart Required. The Apply Changes page displays.
9. Click the Restart Proxy Server button to apply the changes.

Suppressing Outgoing Headers

You can configure the proxy server to remove outgoing headers from the request (usually for security reasons). For example, you might want to prevent the From header from going out because it reveals the user's email address or, you might want to filter out the user-agent header so external servers cannot determine what web browsers your organization uses. You may also want to remove logging or client-related headers that are to be used only in your intranet before a request is forwarded to the Internet.

This feature does not affect headers that are specially handled or generated by the proxy itself or that are necessary to make the protocol work properly (such as If-Modified-Since and Forwarded).

Although it is not possible to stop the forwarded header from originating from a proxy, this is not a security problem. The remote server can detect the connecting proxy host from the connection. In a proxy chain, a forwarded header coming from an inner proxy can be suppressed by an outer proxy. Setting your servers up this way is recommended when you do not want to have the inner proxy or client host name revealed to the remote server.

To suppress outgoing headers

1. Access the Server Manager, and click the Filters tab.
2. Click the Suppress Outgoing Headers link. The Suppress Outgoing Headers page displays.
3. Enter a comma separated list of request headers to be suppressed in the Suppress Headers text box. For example, to suppress the From and User-Agent headers, type **`from,user-agent`**. The headers you type are not case-sensitive. For more information on regular expressions, see "Understanding Regular Expressions" in Chapter 16, "Managing Templates and Resources" on page 349.
4. Click Restart Required. The Apply Changes page displays.
5. Click the Restart Proxy Server button to apply the changes.

Filtering by MIME Type

You can configure the proxy server to block certain files that match a MIME type. For example, you could set up your proxy server to block any executable or binary files so that any clients using your proxy server can't download a possible computer virus.

If you want the proxy server to support a new MIME type, in the Server Manager, choose Preferences > Create/Edit MIME Types and add the type. For more information on creating a MIME type, see "Creating a New MIME Type" on page 140.

You can combine filtering MIME types with templates, so that only certain MIME types are blocked for specific URLs. For example, you could block executables coming from any computer in the `.edu` domain.

To filter by MIME type

1. Access the Server Manager, and click the Filters tab.
2. Click the Set MIME Filters link. The Set MIME Filters page displays.
3. Choose the template you want to use for filtering MIME types, or make sure you are editing the entire server.
4. In the Current filter text box, you can type a regular expression that matches the MIME types you want to block.

   For example, to filter out all applications, you could type **`application/.*`** for the regular expression. This is faster than checking each MIME type for every application type. The regular expression is not case-sensitive. For more information on regular expressions, see "Understanding Regular Expressions" in Chapter 16, "Managing Templates and Resources" on page 349.
5. Check the MIME types you want to filter. When a client attempts to access a file that is blocked, the proxy server returns a "403 Forbidden" message.
6. Click OK.
7. Click Restart Required. The Apply Changes page displays.
8. Click the Restart Proxy Server button to apply the changes.

Filtering by HTML Tags

The proxy server lets you specify HTML tags you want to filter out before passing the file to the client. This lets you filter out objects such as Java applets and JavaScript embedded in the HTML file. To filter HTML tags, you specify the beginning and ending HTML tags. The proxy then substitutes blanks for all text and objects in those tags before sending the file to the client.

NOTE The proxy stores the original (unedited) file in the cache, if the proxy is configured to cache that resource.

To filter out HTML tags

1. Access the Server Manager, and click the Filters tab.
2. Click the Set HTML Tag Filters link. The Set HTML Tag Filters page displays.
3. Choose the template you want to modify. You might choose HTTP, or you might choose a template that specifies only certain URLs such as those from hosts in the `.edu` domain.
4. Check the filter box for any of the default HTML tags you want to filter. The default tags include the following:
    - APPLET usually surrounds Java applets.
    - SCRIPT indicates the start of JavaScript code.
    - IMG specifies an inline image file.
5. You can enter any HTML tags you want to filter. Type the beginning and ending HTML tags.

    For example, to filter out forms, you could type **`FORM`** in the Start Tag box (the HTML tags are not case-sensitive) and **`/FORM`** in the End Tag box. If the tag you want to filter does not have an end tag, such as `OBJECT` and `IMG`, you can leave the End Tag box empty.
6. Click OK.
7. Click Restart Required. The Apply Changes page displays.
8. Click the Restart Proxy Server button to apply the changes.

Configuring the Server for Content Compression

Proxy Server supports HTTP content compression. Content compression allows you to increase delivery speed to clients and serve higher content volumes without increasing your hardware expenses. Content compression reduces content download time, a benefit most apparent to users of dialup and high-traffic connections.

With content compression, your Proxy Server sends out compressed data and instructs the browser to decompress the data on the fly, thus reducing the amount of data sent and increasing page display speed.

Configuring the Server to Compress Content on Demand

You can configure the Proxy Server to compresses transmission data on the fly. A dynamically generated HTML page does not exist until a user asks for it.

To configure your server to compress content on demand

1. Access the Server Manager, and click the Filters tab.
2. Click the Compress Content on Demand link. The Compress Content on Demand page displays.
3. Select the resource from the drop-down list or specify a regular expression. For more information on regular expressions, see "Understanding Regular Expressions" in Chapter 16, "Managing Templates and Resources" on page 349.
4. Specify the following information:
   - **Activate Compress Content on Demand?** Choose whether the server should serve precompressed content for the selected resource.
   - **Vary Header.** Specify whether to insert a Vary: Accept-encoding header. Select either yes or no.If set to yes, then a Vary: Accept-encoding header is always inserted when a compressed version of a file is selected.

     If set to no, then a Vary: Accept-encoding header is never inserted.

     By default, the value is set to yes.
   - **Fragment Size.** Specifies the memory fragment size in bytes to be used by the compression library (`zlib`) to control how much to compress at a time. The default value is 8096.
   - **Compression Level.** Specifies the level of compression. Choose a value between 1 and 9. The value 1 yields the best speed; the value 9 the best compression. The default value is 6, a compromise between speed and compression.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.

7. Click the Restart Proxy Server button to apply the changes.

Chapter 14

Using a Reverse Proxy

This chapter describes how to use Proxy Server as a reverse proxy. Reverse proxy is the name for certain alternate uses of a proxy server. It can be used outside the firewall to represent a secure content server to outside clients, preventing direct, unmonitored access to your server's data from outside your company. It can also be used for replication; that is, multiple proxies can be attached in front of a heavily used server for load balancing. This chapter describes the alternate ways that Proxy Server can be used inside or outside a firewall.

This chapter has the following sections:

- How Reverse Proxying Works
- Setting up a Reverse Proxy

How Reverse Proxying Works

There are two models for reverse proxying. One model takes advantage of Proxy Server's security features to handle transactions, and the other makes use of its caching features to provide load balancing on a heavily used server. Both of these models differ from the conventional proxy usage in that they do not operate strictly on a firewall.

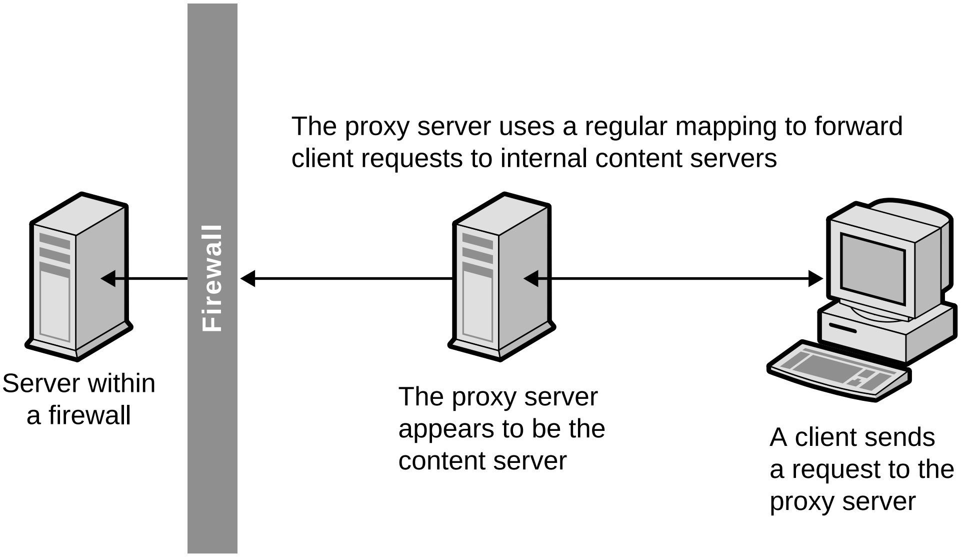
Proxy as a Stand-in for a Server

If you have a content server that has sensitive information that must remain secure, such as a database of credit card numbers, you can set up a proxy outside the firewall as a stand–in for your content server. When outside clients try to access the content server, they are sent to the proxy server instead. The real content resides on your content server, safely inside the firewall. The proxy server resides outside the firewall, and appears to the client to be the content server.

When a client makes a request to your site, the request goes to the proxy server. The proxy server then sends the client's request through a specific passage in the firewall to the content server. The content server passes the result through the passage back to the proxy. The proxy sends the retrieved information to the client, as if the proxy were the actual content server (see Figure 14-1). If the content server returns an error message, the proxy server can intercept the message and change any URLs listed in the headers before sending the message to the client. This prevents external clients from getting redirection URLs to the internal content server.

In this way, the proxy provides an additional barrier between the secure database and the possibility of malicious attack. In the unlikely event of a successful attack, the perpetrator is more likely to be restricted to only the information involved in a single transaction, as opposed to having access to the entire database. The unauthorized user can not get to the real content server because the firewall passage allows only the proxy server to have access.

Figure 14-1 A reverse proxy appears to be the real content server


Firewall
The proxy server uses a regular mapping to forward client requests to internal content servers
Server within a firewall
The proxy server appears to be the content server
A client sends a request to the proxy server

You can configure the firewall router to allow a specific server on a specific port (in this case, the proxy on its assigned port) to have access through the firewall without allowing any other machines in or out.

Secure Reverse Proxying

Secure reverse proxying occurs when one or more of the connections between the proxy server and another machine uses the Secure Sockets Layer (SSL) protocol to encrypt data.

Secure reverse proxying has many uses:

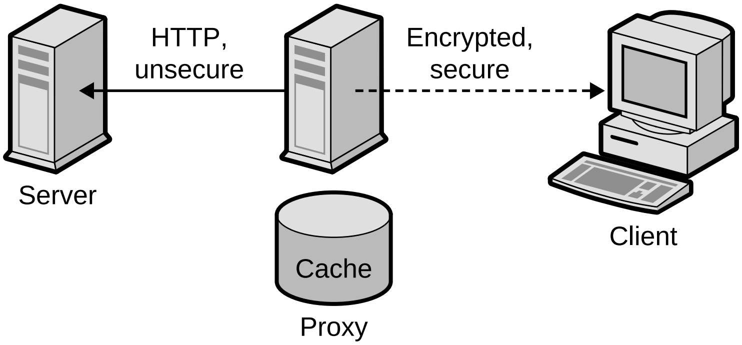
- It can provide an encrypted connection from a proxy server outside a firewall to a secure content server inside the firewall.
- It can allow clients to connect securely to the proxy server, facilitating the secure transmission of information (such as credit card numbers).

Secure reverse proxying causes each secure connection to be slower due to the overhead involved in encrypting your data. However, because SSL provides a caching mechanism, two connecting parties can reuse previously negotiated security parameters, dramatically reducing the overhead on subsequent connections.

There are three ways to configure a secure reverse proxy:

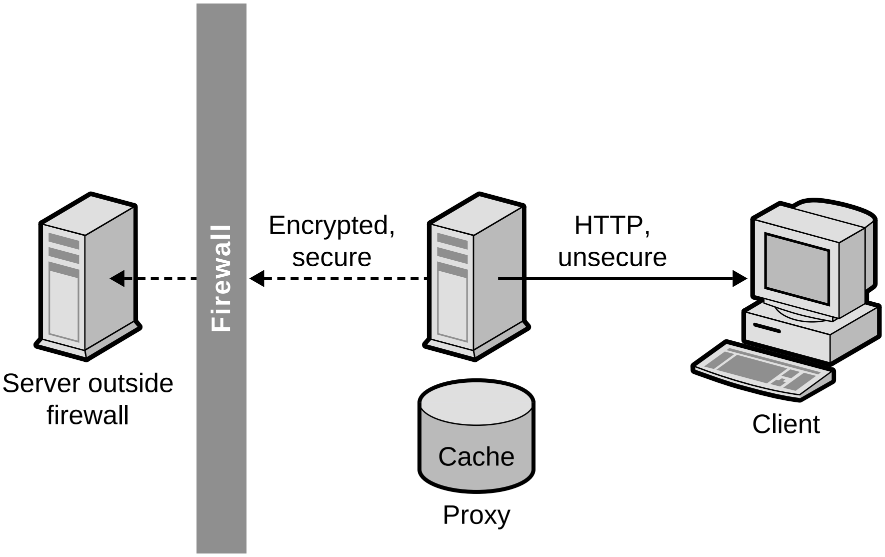
Secure client to proxy. This scenario is effective if there is little or no chance that the information being exchanged between your proxy and content server can be accessed by unauthorized users (see Figure 14-2).

Figure 14-2 Secure client connection to proxy



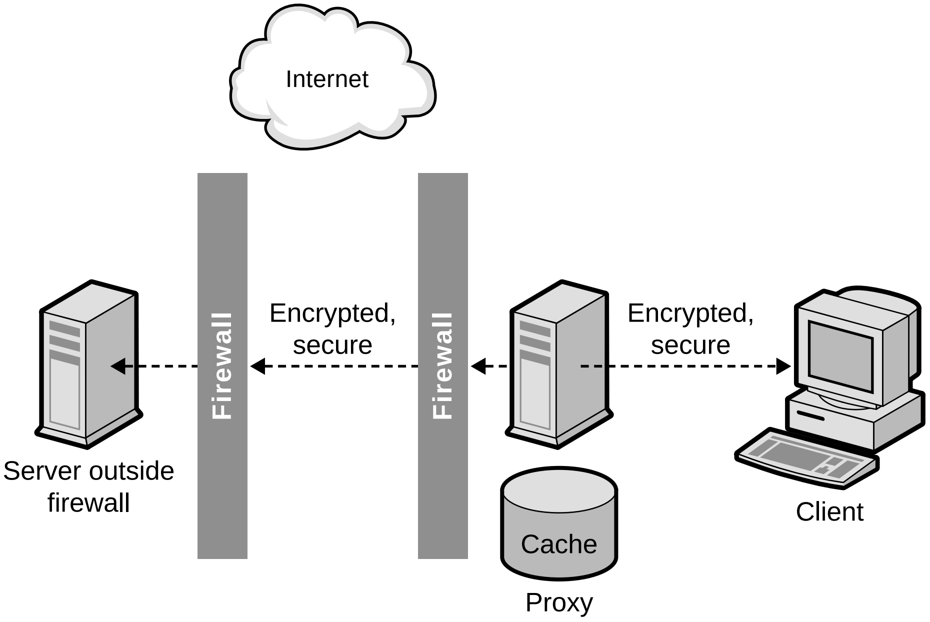
Secure proxy to content server. This scenario is effective if you have clients inside the firewall and a content server that is outside the firewall. In this scenario, your proxy server can act as a secure channel between sites (see Figure 14-3

Figure 14-3 Secure proxy connection to content server



- **Secure client to proxy and secure proxy to content server**. This scenario is effective if the information exchanged between the server, proxy and client needs to be secure. In this scenario, your proxy server can act like a secure channel between sites with the additional security of client authentication (see Figure 14-4).

Figure 14-4 Secure client connection to proxy and secure proxy connection to content server



For information on how to set up each of these configurations, see "Setting up a Reverse Proxy" on page 321.

In addition to SSL, the proxy can use client authentication, which requires that a computer making a request to the proxy provides a certificate (or form of identification) to verify its identity.

Proxying for Load Balancing

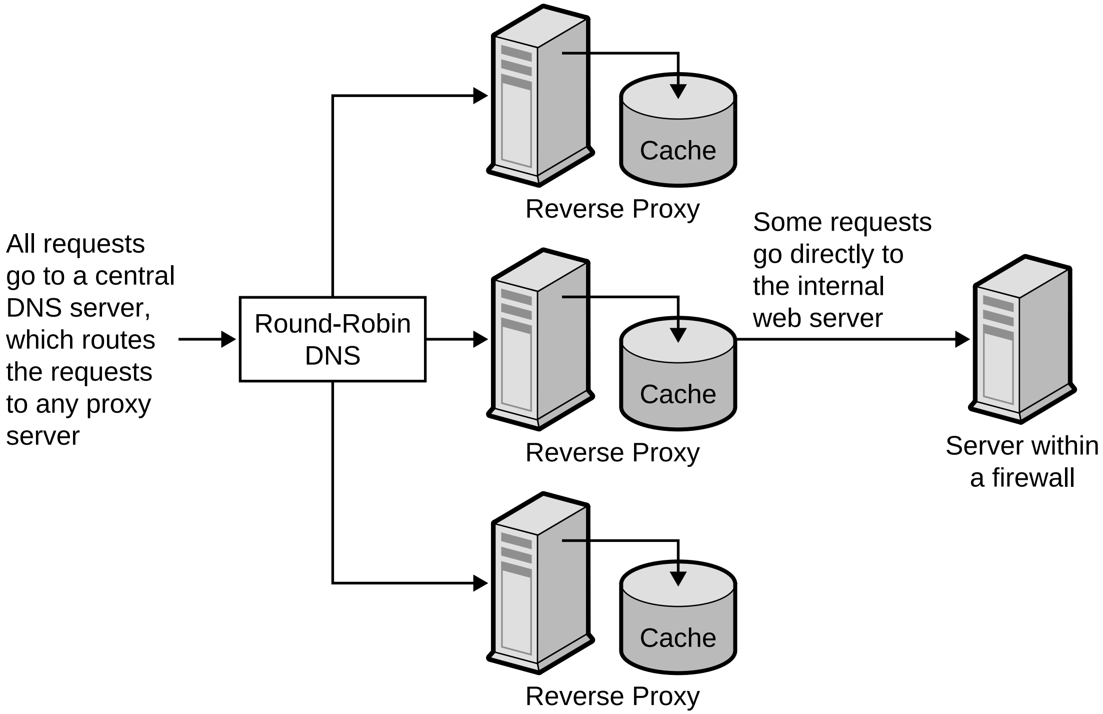
You can use multiple proxy servers within an organization to balance the network load among web servers. This model lets you take advantage of the caching features of the proxy server to create a server pool for load balancing. In this case, the proxy servers can be on either side of the firewall. If you have a web server that receives a high number of requests per day, you could use proxy servers to take the load off the web server and make the network access more efficient.

The proxy servers act as go-betweens for client requests to the real server. The proxy servers cache the requested documents. If there is more than one proxy server, DNS can route the requests randomly using a “round-robin” selection of their IP addresses. The client uses the same URL each time, but the route the request takes might go through a different proxy each time.

The advantage of using multiple proxies to handle requests to one heavily used content server is that the server can handle a heavier load, and more efficiently than it could alone. After an initial start-up period in which the proxies retrieve documents from the content server for the first time, the number of requests to the content server can drop dramatically.

Only CGI requests and occasional new requests must go all the way to the content server. The rest can be handled by a proxy. Here is an example. Suppose that 90% of the requests to your server are not CGI requests (which means they can be cached), and that your content server receives 2 million hits per day. In this situation, if you connect three reverse proxies, and each of them handles 2 million hits per day, about 6 million hits per day would then be possible. The 10% of requests that reach the content server could add up to about 200,000 hits from each proxy per day, or only 600,000 total, which is far more efficient. The number of hits could increase from around 2 million to 6 million, and the load on the content server could decrease correspondingly from 2 million to 600,000. Your actual results would depend upon your situation.

Figure 14-5 Proxy used for load balancing


All requests go to a central DNS server, which routes the requests to any proxy server
Round-Robin DNS
Cache
Reverse Proxy
Cache
Reverse Proxy
Cache
Reverse Proxy
Some requests go directly to the internal web server
Server within a firewall

Setting up a Reverse Proxy

To set up a reverse proxy, you need two mappings: a regular and a reverse mapping.

- The regular mapping redirects requests to the content server. When a client requests a document from the proxy server, the proxy server needs a regular mapping to tell it where to get the actual document.

CAUTION You should not use a reverse proxy with a proxy that serves autoconfiguration files. This is because the proxy could return the wrong result.

- The reverse mapping makes the proxy server trap for redirects from the content server. The proxy intercepts the redirect and then changes the redirected URL to map to the proxy server. For example, if the client requests a document that was moved or not found, the content server will return a message to the client explaining that it cannot find the document at the requested URL. In that returned message, the content server adds an HTTP header that lists a URL to use to get the moved file. In order to maintain the privacy of the internal content server, the proxy can redirect the URL using a reverse mapping.

Suppose you have a web server called `http://http.site.com/` and you want to set up a reverse proxy server for it. You could call the reverse proxy `http://proxy.site.com/`.

You would create a *regular* mapping and a *reverse* mapping as follows:

1. Access the Server Manager, and click the URLs tab.
2. Click the Create Mapping link. The Create Mapping page displays.
3. In the page that appears, enter information for a single mapping. For example:

   Regular mapping:

   Source prefix: `http://proxy.site.com`

   Source destination: `http://http.site.com/`

4. Click OK. Return to the page and create the second mapping:

   Reverse mapping:

   Source prefix: `http://http.site.com/`

   Source destination: `http://proxy.site.com/`

5. To make the change, click OK.

   Once you click the OK button, the proxy server adds one or more additional mappings. To see the mappings, click the link called View/Edit Mappings. Additional mappings would be in the following format:

   from: `/`

   to: `http://http.site.com/`

   These additional automatic mappings are for users who connect to the reverse proxy as a normal server. The first mapping is to catch users connecting to the reverse proxy as a regular proxy. Depending on the setup, usually the second is the only one required, but it does not cause problems in the proxy to have them both.

NOTE If the web server has several DNS aliases, each alias should have a corresponding regular mapping. If the web server generates redirects with several DNS aliases to itself, each of those aliases should have a corresponding reverse mapping.

CGI applications still run on the origin server; the proxy server never runs CGI applications on its own. However, if the CGI script indicates that the result can be cached (by implying a non-zero time-to-live by issuing a Last-modified or Expires header), the proxy will cache the result.

CAUTION When authoring content for the web server, keep in mind that the content will be served by the reverse proxy, too, so all links to files on the web server should be relative links. There must be *no* reference to the host name in the HTML files; that is, all links must be of the page:

```
/abc/def
```

as opposed to a fully qualified host name, such as:

```
http://http.site.com/abc/def
```

Setting up a Secure Reverse Proxy

Before setting up secure reverse proxying, you should be familiar with digital certificates, Certificate Authorities, and authentication.

Setting up a secure reverse proxy is almost the same as setting up an insecure reverse proxy. The only difference is that you need to specify HTTPS as the protocol for the files to be encrypted.

The following instructions explain how to set up your secure reverse proxy according to the configuration scenario you choose. To demonstrate how to set up mappings, the instructions suppose that you have a web server called `http.site.com` and that you want to set up a secure reverse proxy server called `proxy.site.com`. When following the steps, substitute the name of your web server and proxy for the example names used in the directions.

Secure Client to Proxy

1. Access the Server Manager, and click the URLs tab.
2. Click the Create Mapping link. The Create Mapping page displays.
3. In the page that appears, set up regular and reverse mappings in the following manner:

   Regular mapping:

   Source prefix: `https://proxy.mysite.com`

   Source destination: `http://http.mysite.com/`

   Reverse mapping:

   Source prefix: `http://http.mysite.com/`

   Source destination: `https://proxy.mysite.com/`
4. Save and apply your changes.

   To see the mappings you just created, click the link called View/Edit Mappings.

NOTE This configuration will only work if your proxy server is running in secure mode. In other words, encryption must be enabled and the proxy must be restarted from the command line. To restart the proxy from the command line, go to the proxy directory and type **`./start`**.

Secure Proxy to Content Server

1. Access the Server Manager, and click the URLs tab.
2. Click the Create Mapping link. The Create Mapping page displays.

3. In the page that appears, set up regular and reverse mappings in the following manner:

   Regular mapping:

   Source prefix: `http://proxy.mysite.com`

   Source destination: `https://http.mysite.com/`

   Reverse mapping:

   Source prefix: `https://http.mysite.com/`

   Source destination: `http://proxy.mysite.com/`

4. Save and Apply your changes. To see the mappings you just created, click the link called View/Edit Mappings.

NOTE This configuration will only work if your content server is running in secure mode.

Secure Client to Proxy and Secure Proxy to Content Server

1. Access the Server Manager, and click the URLs tab.
2. Click the Create Mapping link. The Create Mapping page displays.
3. In the page that appears, set up regular and reverse mappings in the following manner:

   Regular mapping:

   Source prefix: `https://proxy.mysite.com`

   Source destination: `https://http.mysite.com/`

   Reverse mapping:

   Source prefix: `https://http.mysite.com/`

   Source destination: `https://proxy.mysite.com/`

4. Save and Apply your changes. To see the mappings you just created, click the link called View / Edit Mappings.

NOTE	This configuration will only work if your proxy server and content server are running in secure mode. In other words, for the proxy, encryption must be enabled and the proxy must be restarted from the command line. To restart the proxy from the command line, go to the proxy directory and type **`./restart`**.

Virtual Multihosting in Reverse Proxy

Virtual multihosting is a feature which allows an origin server, or in our case, a reverse proxy server, to respond to multiple DNS aliases as if there was a different server installed in each of those addresses. As an example, you could have the DNS host names:

- `www`
- `specs`
- `phones`

Each of them could be mapped to the same IP address (the IP address of the reverse proxy). You could then have the reverse proxy act differently based on which DNS name was used to access it.

Virtual Multihosting allows you to host multiple different *domains* in a single reverse proxy server as well. For example:

- `www.domain-1.com`
- `www.domain-2.com`
- `www.domain-3.com`

Note that you can have a combination of multiple local host names as well as multiple domains, all in a single proxy server:

- `www`
- `specs`
- `phones`
- `www.domain-1.com`
- `www.domain-2.com`

- `www.domain-3.com`

This section contains the following topics:

- Functional Details of Virtual Multihosting
- Important Notes on Virtual Multihosting

Functional Details of Virtual Multihosting

The virtual multihosting feature works by specifying the DNS host and domain names (or aliases), and then giving a target URL prefix where requests sent to that host name should be directed. As an example, you can have two mappings:

- `engr.domain.com -> http://int-engr.domain.com`
- `mktg.domain.com -> http://int-mktg.domain.com`

Mappings do not have to go root-to-root; you may specify an additional URL path prefix in the target URL:

- `engr.domain.com -> http://internal.domain.com/engr`
- `mktg.domain.com -> http://internal.domain.com/mktg`

Same applies to virtual domain mappings. For example, you could use:

- `www.domain-1.com -> http://int-engr.domain.com`
- `www.domain-2.com -> http://int-mktg.domain.com`

The system will look at the HTTP "Host:" header, and based on that header, it will choose the matching Virtual Multihosting mapping. If none of the multihosting mappings match, the server will continue looking at other mappings in the order that they appear in the configuration file, or perform no mappings if no matches are found. If there are no matches, the proxy will typically respond with the "Proxy denies fulfilling the request" response.

To configure virtual multihosting

1. Access the Server Manager and click the URLs tab.
2. Click the Configure Virtual Multihosting link. The Configure Virtual Multihosting page displays.
3. In the Source Hostname (alias) field, specify the local host name (or DNS alias) that this mapping should apply to.
4. In the Source Domain Name field, enter the local domain name that this mapping should apply to. Typically, this is your own network’s domain name, unless you want to multi-host multiple different DNS domains.

5. In the Destination URL Prefix field, enter the target URL prefix where the request will be directed if the host and domain names match the above specifications.

6. If you are using templates, choose the template name from the Use This Template drop-down list, or leave the value at NONE" if you do not want to apply a template.

7. Click OK.

8. Click Restart Required. The Apply Changes page displays.

9. Click the Restart Proxy Server button to apply the changes.

   Repeat the above steps for each virtual multihosting mapping you want to establish.

All virtual multihosting mappings appear on the bottom of the Configure Virtual Multihosting page. Note that the Source Hostname (alias) and Source Domain Name fields are merged, together with the proxy's port number, into a single regular expression that is used to match the "Host:" header.

For example, if you have host name "www", domain "example.com", and port number "8080", it will display the regular expression:

```
www(|.example.com)(|:8080)
```

This will guarantee a match with all of the following possible combinations that the user may have typed, or the client may have sent (the port number may be omitted by some client software even when it's non-80, as it is obvious to the server which port number it was listening on):

- `www`
- `www:8080`
- `www.example.com`
- `www.example.com:8080`

Important Notes on Virtual Multihosting

You will need to disable the Client autoconfiguration feature before you can configure reverse proxy mappings. Doing so will not cause any problems because the Client autoconfiguration feature is for the forward proxy operation, not reverse proxy.

The Virtual Multihosting feature establishes automatic reverse mappings. In other words, do not create reverse mappings for mappings that you enter using the Virtual Multihosting page.

Virtual mappings are specified with `virt-map` function in `obj.conf`.

Virtual mappings are matched in the order specified in the obj.conf configuration file. If there are regular, reverse, regular expression, or client autoconfiguration mappings before the virtual mappings, they will be applied first. Similarly, if no matches are found in virtual mappings, translation will continue to the next mapping after the virtual mapping section in `obj.conf`.

If the port number of the proxy server is changed, you will need to recreate the Virtual Multihosting mappings, as they now have the wrong port number.

Chapter 15

Using SOCKS

This chapter describes how to configure and use the SOCKS server included with Sun Java System Web Proxy Server. Proxy Server supports SOCKS versions 4 and 5.

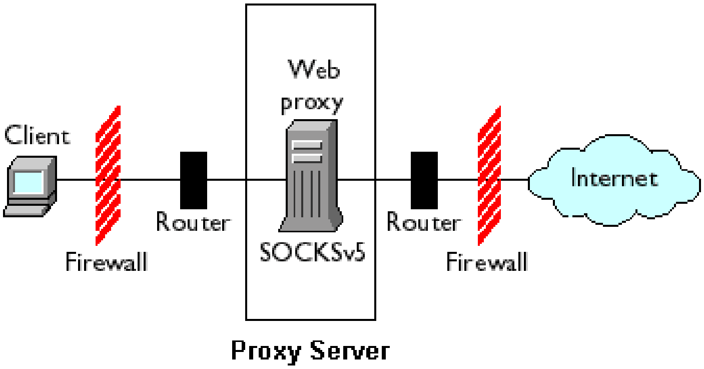
This chapter contains the following sections:

About SOCKS

SOCKS is a networking proxy protocol that redirects connection requests from hosts on opposite sides of a SOCKS server, enabling hosts on one side to gain full access to hosts on the other without requiring direct IP reachability. SOCKS is commonly used as a network firewall that enables hosts behind a SOCKS server to gain full access to the Internet, while preventing unauthorized access from the Internet to the internal hosts.

A SOCKS server is a generic firewall daemon that controls access through the firewall on a point-to-point basis. The SOCKS server authenticates and authorizes requests, establishes a proxy connection, and relays data. The SOCKS server works at the network level instead of the application level, and therefore has no knowledge of protocols or methods used for transferring requests. Because the SOCKS server has no knowledge of protocols, it can be used to pass those protocols that are not supported by the Proxy Server (such as Telnet).

Figure 15-1 Position of a SOCKS Server in a Network



Using the Bundled SOCKS v5 Server

Sun Java System Web Proxy Server includes its own SOCKS daemon that understands the standard `socks5.conf` file format used by other SOCKS daemons. This daemon can be used by the Proxy Server to route requests, or it can run standalone from the Proxy Server to provide additional capabilities for the network. For more information about configuring the Proxy Server to route requests through a SOCKS server, see "Configuring Routing Entries" on page 344.

The SOCKS daemon included with Proxy Server is disabled by default, and can be enabled from the SOCKS tab in the Server Manager interface, or from the command line. For more information, see "Starting and Stopping the SOCKS v5 Server" on page 336.

NOTE	In Proxy Server 4 the name of the SOCKS daemon has been changed from `ns-sockd` to `sockd`.

Following are the high-level steps that must be taken to use the SOCKS server included with the Proxy Server:

1. Configure the SOCKS server (see "Configuring the SOCKS v5 Server" on page 337).
2. If the SOCKS server will be running on a computer with multiple interfaces, create SOCKS routing entries (see "Configuring Routing Entries" on page 344).
3. Create authentication entries (see "Configuring SOCKS v5 Authentication Entries" on page 338).
4. Create connection entries (see "Configuring SOCKS v5 Connection Entries" on page 341).
5. Enable the SOCKS server (see "Starting and Stopping the SOCKS v5 Server" on page 336).

About socks5.conf

Sun Java System Web Proxy Server uses the `socks5.conf` file to control access to the SOCKS server and its services. Each line defines what the Proxy Server does when a request is received that matches the line. Choices made in the Server Manager are written to `socks5.conf`. The file can also be edited manually. The `socks5.conf` file is located in the installation root directory (*server_root*) as follows:

server_root`/proxy-`*serverid*`/config` directory

This section provides general information about `socks5.conf`. For detailed information about the file and its directives and syntax, see the Proxy Server *Configuration File Reference*.

Authentication

The SOCKS daemon can be configured to require authentication to use its services. Authentication is based on the host name and port of the connecting client. If you choose to require a user name and password, the information is authenticated against a user name and password file referenced by the `socks5.conf` file. If the provided user name and password do not match a listing in the password file, access is denied. The format for user names and passwords in the password file is *username password*, where the user name and password are separated by a space. You can also ban users. To require user name and password authentication, the `SOCKS5_PWDFILE` directive must be added to `socks5.conf`. For more information about the directive and its syntax, see the `socks5.conf` section in the Proxy Server's *Configuration File Reference*.

User name and password authentication can also be performed against a configured LDAP server, and not just a file.

Access Control

Access control is performed using a set of ordered lines in the `socks5.conf` file. Each line contains a single directive that permits or denies access to a resource. Directives are processed in the order in which they appear in the configuration file. A request that does not match any of the permit directives is denied access.

Logging

The SOCKS daemon logs both error and access messages in the SOCKS log file. The log file location and type of logging can be specified in `socks5.conf`.

The SOCKS daemon also generates a stat entry each hour, which gives statistics for the daemon.

Tuning

You can use the `socks5.conf` file to determine the number of worker and accept threads used by the SOCKS server. These numbers influence the performance of the SOCKS server.

For more information about worker and accept thread settings and their impact on performance, see the relevant section in "Configuring the SOCKS v5 Server" on page 337.

Starting and Stopping the SOCKS v5 Server

The SOCKS server can be started and stopped from the Server Manager or from the command line.

To start and stop the SOCKS server from the Server Manager

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Start / Stop SOCKS Server link.
3. Start or stop the SOCKS server.

To start and stop the SOCKS server from the command line

Run the scripts found in the *server_root*/`proxy-`*serverid* directory, where *server_root* is the installation root:

- `start-sockd` starts the SOCKS daemon

- `stop-sockd` stops the SOCKS daemon
- `restart-sockd` restarts the SOCKS daemon

Configuring the SOCKS v5 Server

To configure the SOCKS server

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Configure SOCKS v5 link.
3. In the SOCKS Port field, enter the port number on which the SOCKS server will listen (1080 by default).
4. Select the SOCKS options you want to use. The following options are available:
    - **Disable Reverse DNS Lookup.** Disables reverse DNS lookup for the SOCKS server. Reverse DNS translates IP addresses into host names. Disabling reverse DNS lookup can conserve network resources. This is disabled by default (that is, the Disable Reverse DNS Lookup checkbox is selected by default). If reverse DNS lookup is disabled, and a URL is requested with a host name, the server will not map the host name to the IP address. If reverse DNS lookup is enabled, the server performs the mapping, and an entry is added to the SOCKS log file, listing the DNS translation.
    - **Use Client-specific Bind Port.** Allows the client to specify the port in a BIND request. With this option disabled, SOCKS ignores the client's requested port and assigns a random port. This is disabled by default.
    - **Allow Wildcard As Bind IP Address.** Allows the client to specify an IP address of all zeros (0.0.0.0) in a BIND request, which means that any IP address can connect. With this option disabled, the client must specify the IP address that will be connecting to the bind port, and the SOCKS server rejects requests to bind to 0.0.0.0. This is disabled by default.
    - **Quench Updates.** Disables the automatic stat file writing once an hour. If disabled, the writing takes place with every request (see "Logging" on page 336).

NOTE The Quench Updates element displays in the user interface but is not implemented in this release of Proxy Server 4.

5. In the Log File field, enter the full path name of the SOCKS log file. The default is *server_root*/`proxy-`*serverid*/`logs/socks5.log`.

6. From the Log Level drop-down list, select whether the log file should contain warnings and errors only, all requests, or debugging messages.

7. Select an RFC 1413 ident response. Ident allows the SOCKS server to determine the user name for a client. Generally, this feature only works when the client is running some flavor of UNIX. The following options are available

   - **Don't Ask.** Never use ident to determine the user name for a client. This is the recommended and default setting.

   - **Ask But Don't Require.** Ask for the user name of all clients, but do not require it. This option uses ident for logging purposes only.

   - **Require.** Ask for the user name of all clients, and only permit access to those with valid responses.

8. In the SOCKS Tuning section, specify the number of worker and accept threads the SOCKS server should use (these numbers influence performance of the SOCKS server), and then click OK:

   - **Number Of Worker Threads.** The default is 40. If the SOCKS server is too slow, increase the number of worker threads. If it is unstable, decrease the number. When changing this number, start with the default and increase or decrease as necessary. The typical number of worker threads is between 10 and 150. The absolute maximum is 512, but having more than 150 tends to be wasteful and unstable.

   - **Number Of Posted Accepts.** The default is 1. If the SOCKS server is dropping connections, increase the number of accept threads. If it is unstable, decrease the number. When changing this number, start with the default and increase or decrease as necessary. The typical number of accept threads is between 1 and 10. The absolute maximum is 512, but having more than 60 tends to be wasteful and unstable. This is a very important setting. Tune this setting if requests are failing when the SOCKS server is put under load and connections are being dropped.

Configuring SOCKS v5 Authentication Entries

SOCKS authentication entries identify the hosts from which the SOCKS daemon should accept connections, and which types of authentication the SOCKS daemon should use to authenticate those hosts.

This section contains the following topics:

- Creating Authentication Entries
- Editing Authentication Entries
- Deleting Authentication Entries
- Moving Authentication Entries

Creating Authentication Entries

To create SOCKS authentication entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Authentication link.
3. Click the Add button.
4. In the Host Mask field, enter the IP addresses or host names of the hosts that the SOCKS server will authenticate. If you enter an IP address, follow the address with a forward slash and the mask to be applied to the incoming IP address. The SOCKS server applies this mask to the IP address to determine if it is a valid host. Do not use spaces in the host mask entry. If you do not enter a host mask, the authentication entry applies to all hosts.

   For example, you can enter 155.25.0.0/255.255.0.0 in the host mask field. If the host's IP address is 155.25.3.5, the SOCKS server applies the mask to the IP address and determines that the host's IP address matches the IP address for which the authentication record applies (155.25.0.0).

5. In the Port Range field, enter the ports on the host computers that the SOCKS server will authenticate. Do not use spaces in the port range entry. If you do not enter a port range, the authentication entry applies to all ports.

   You can use brackets [ ] to include the ports at each end of the range or parentheses ( ) to exclude them. For example, [1000-1010] means all port numbers between and including 1000 and 1010, while (1000-1010) means all port numbers between, but not including, 1000 and 1010. You can also mix brackets and parentheses. For instance, (1000-1010] means all numbers between 1000 and 1010, excluding 1000, but including 1010.

6. From the Authentication Type drop-down list, select the authentication type. The following options are available

- **Require user-password.** User name and password are required to access the SOCKS server.
- **User-password, if available.** If a user name and password are available, they should be used to access the SOCKS server (but they are not required for access).
- **Ban.** Banned from the SOCKS server.
- **None.** No authentication is required to access the SOCKS server.

7. From the Insert drop-down list, select the position for this entry in the `socks5.conf` file and click OK. Because you can have multiple authentication methods, you must specify the order in which they are evaluated. Therefore, if the client does not support the first authentication method listed, the second method is used instead. If the client does not support any of the authentication methods listed, the SOCKS server disconnects without accepting a request.

Editing Authentication Entries

To edit authentication entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Authentication link.
3. Select the authentication entry you want to edit and click the Edit button.
4. Make changes as desired and click OK.

Deleting Authentication Entries

To delete authentication entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Authentication link.
3. Select the authentication entry you want to delete and click the Delete button.

Moving Authentication Entries

Entries are evaluated in the order in which they appear in the `socks5.conf` file. You can change the order by moving them.

To move authentication entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Authentication link.
3. Select the authentication entry you want to move and click the Move button.
4. From the Move drop-down list, select the position for this entry in the `socks5.conf` file and click OK.

Configuring SOCKS v5 Connection Entries

SOCKS connection entries specify whether the SOCKS daemon should permit or deny a request.

This section contains the following topics:

- Creating Connection Entries
- Editing Connection Entries
- Deleting Connection Entries
- Moving Connection Entries

Creating Connection Entries

To create connection entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Connections link.
3. Click the Add button.
4. From the Authentication Type drop-down list, select the authentication method for which this access control line applies.
5. From the Connection Type drop-down list, select the type of command the line matches. Possible command types are:
   - **Connect**
   - **Bind**
   - **UDP**

- **All**

6. In the Source Host Mask field, enter the IP address or host names of the hosts for which the connection control entry applies. If you enter an IP address, follow it with a forward slash and the mask to be applied to the source's IP address. The SOCKS server applies this mask to the source's IP address to determine if it is a valid host. Do not use spaces in the host mask entry. If you do not enter a host mask, the connection entry applies to all hosts.

   For example, you can enter 155.25.0.0/255.255.0.0 in the host mask field. If the host's IP address is 155.25.3.5, the SOCKS server applies the mask to the IP address and determines that the host's IP address matches the IP address for which the connection control entry applies (155.25.0.0).

7. In the Port Range field, enter the ports on the source computers for which the connection control entry applies. Do not use spaces in the port range entry. If you do not specify a port range, the connection entry applies to all ports.

   You can use brackets [ ] to include the ports at each end of the range or parentheses ( ) to exclude them. For example, [1000-1010] means all port numbers between and including 1000 and 1010, while (1000-1010) means all port numbers between, but not including, 1000 and 1010. You can also mix brackets and parentheses. For instance, (1000-1010] means all numbers between 1000 and 1010, excluding 1000, but including 1010.

8. In the Destination Host Mask field, enter the IP address or host name for which the connection entry applies. If you enter an IP address, follow it with a forward slash and the mask to be applied to the incoming IP address. The SOCKS server applies this mask to the IP address of the destination computer to determine if it is a valid destination host. Do not use spaces in the host mask entry. If you do not enter a destination host mask, the connection entry applies to all hosts.

   For example, you can enter 155.25.0.0/255.255.0.0 into the destination host mask field. If the destination host's IP address is 155.25.3.5, the SOCKS server applies the mask to the IP address and determines that the destination host's IP address matches the IP address for which the proxy entry applies (155.25.0.0).

9. In the Port Range field, enter the ports on the destination host computers for which the connection control entry applies. Do not use spaces in the port range entry. If you do not enter a port range, the connection entry applies to all ports.

NOTE Most SOCKS applications request port 0 for bind requests, meaning they have no port preference. Therefore, the destination port range for bind should always include port 0.

You can use brackets [ ] to include the ports at each end of the range or parentheses ( ) to exclude them. For example, [1000-1010] means all port numbers between and including 1000 and 1010, while (1000-1010) means all port numbers between, but not including, 1000 and 1010. You can also mix brackets and parentheses. For instance, (1000-1010] means all numbers between 1000 and 1010, excluding 1000, but including 1010.

10. In the User Group field, enter the group to which you want to permit or deny access. If a group is not specified, the connection entry applies to all users.
11. From the Action drop-down list, choose to permit or deny access for the connection you are creating.
12. From the Insert drop-down list, select the position for this entry in the `socks5.conf` file and click OK. Because you can have multiple connection directives, you must specify the order in which they are evaluated.

Editing Connection Entries

To edit connection entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Connections link.
3. Select the connection entry you want to edit and click the Edit button.
4. Make changes as desired and click OK.

Deleting Connection Entries

To delete connection entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Connections link.
3. Select the connection entry you want to delete and click the Delete button.

Moving Connection Entries

Entries are evaluated in the order in which they appear in the `socks5.conf` file. You can change the order by moving them.

To move connection entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Connections link.
3. Select the connection entry you want to move and click the Move button.
4. From the Move drop-down list, select the position for this entry in the `socks5.conf` file and click OK.

Configuring SOCKS v5 Server Chaining

SOCKS servers can be chained together in the same manner as Proxy Servers, meaning that a SOCKS server can route through another SOCKS server.

To configure SOCKS server chaining

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Routing link.
3. In the Server Chaining section, enter the user name and password for authenticating to chained Proxy Servers (if the downstream proxy in the proxy chain requires authentication to serve any requests), and then click OK.

Configuring Routing Entries

Routing entries can be used to configure a Proxy Server to route requests through a SOCKS server. There are two types of routing entries, the SOCKS v5 routes and the SOCKS v5 proxy routes:

- The SOCKS v5 routes identify which interface the SOCKS daemon should use for particular IP addresses.
- The SOCKS v5 proxy routes identify the IP addresses that are accessible through another SOCKS server, and whether that SOCKS server connects directly to the host. Proxy routes are important when routing through a SOCKS server.

This section contains the following topics:

- Creating SOCKS v5 Routing Entries
- Creating SOCKS v5 Proxy Routing Entries
- Editing Routing Entries
- Deleting Routing Entries
- Moving Routing Entries

Creating SOCKS v5 Routing Entries

To create routing entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Routing link.
3. In the Routing section, click the Add button.
4. In the Host Mask field, enter the IP address or host name for which incoming and outgoing connections must go through the specified interface. If you enter an IP address, follow it with a forward slash and the mask to be applied to the incoming IP address. The SOCKS server applies this mask to the IP address to determine if it is a valid host. Do not use spaces in the host mask entry. If you do not enter a host mask, the SOCKS v5 entry applies to all hosts.

   For example, you can enter 155.25.0.0/255.255.0.0 in the host mask field. If the host's IP address is 155.25.3.5, the SOCKS server applies the mask to the IP address and determines that the host's IP address matches the IP address for which the routing entry applies (155.25.0.0).
5. In the Port Range field, enter the ports for which incoming and outgoing connections must go through the specified interface. Your port range should not have any spaces. If you do not specify a port range, the SOCKS v5 entry applies to all ports.

   You can use brackets [ ] to include the ports at each end of the range or parentheses ( ) to exclude them. For example, [1000-1010] means all port numbers between and including 1000 and 1010, while (1000-1010) means all port numbers between, but not including, 1000 and 1010. You can also mix brackets and parentheses. For instance, (1000-1010] means all numbers between 1000 and 1010, excluding 1000, but including 1010.

6. In the Interface/Address field, enter the IP address or name of the interface through which incoming and outgoing connections must pass.

7. From the Insert drop-down list, select the position for this entry in the `socks5.conf` file and click OK. Because you can have multiple routing methods, you must specify the order in which they are evaluated.

NOTE	The interface specified should be used for both incoming and outgoing connections, otherwise the incoming route will be different from the configured interface and an error message will be received.

Creating SOCKS v5 Proxy Routing Entries

To create proxy routing entries

1. Access the Server Manager for a server instance and click the SOCKS tab.

2. Click the Set SOCKS v5 Routing link.

3. In the Proxy Routing section, click the Add button.

4. From the Proxy Type drop-down list, select the type of Proxy Server through which you are routing. The following options are available

   - **SOCKS v5**
   - **SOCKS v4**
   - **Direct connection**

5. In the Destination Host Mask field, enter the IP address or host name for which the connection entry applies. If you enter an IP address, follow it with a forward slash and the mask to be applied to the incoming IP address. The SOCKS server applies this mask to the IP address of the destination computer to determine if it is a valid destination host. Do not use spaces in the host mask entry. If you do not enter a destination host mask, the connection entry applies to all hosts.

   For example, you can enter 155.25.0.0/255.255.0.0 in the destination host mask field. If the destination host's IP address is 155.25.3.5, the SOCKS server applies the mask to the IP address and determines that the destination host's IP address matches the IP address for which the proxy entry applies (155.25.0.0).

6. In the Destination Port Range field, enter the ports on the destination host for which the proxy entry applies. Do not use spaces in the port range entry. If you do not specify a port range, the proxy entry applies to all ports.

   You can use brackets [ ] to include the ports at each end of the range or parentheses ( ) to exclude them. For example, [1000-1010] means all port numbers between and including 1000 and 1010, while (1000-1010) means all port numbers between, but not including, 1000 and 1010. You can also mix brackets and parentheses. For instance, (1000-1010] means all numbers between 1000 and 1010, excluding 1000, but including 1010.

7. In the Destination Proxy Address field, enter the host name or IP address of the Proxy Server to use.

8. In the Destination Proxy Port field, enter the port number on which the Proxy Server will listen for SOCKS requests.

9. From the Insert drop-down list, select the position for this entry in the `socks5.conf` file and click OK. Because you can have multiple routing methods, you must specify the order in which they are evaluated.

Editing Routing Entries

To edit routing entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Routing link.
3. Select the entry you want to edit and click the Edit button.
4. Make changes as desired and click OK.

Deleting Routing Entries

To delete routing entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Routing link.
3. Select the entry you want to delete and click the Delete button.

Moving Routing Entries

Entries are evaluated in the order in which they appear in the `socks5.conf` file. You can change the order by moving them.

To move routing entries

1. Access the Server Manager for a server instance and click the SOCKS tab.
2. Click the Set SOCKS v5 Routing link.
3. Select the entry you want to move and click the Move button.
4. From the Move drop-down list, select the position for this entry in the `socks5.conf` file and click OK.

Chapter 16

Managing Templates and Resources

Templates allow you to group URLs together so that you can configure how the proxy handles them. You can make the proxy behave differently depending on the URL the client tries to retrieve. For example, you might require the client to authenticate (type in a user name and password) when accessing URLs from a specific domain. Or, you might deny access to URLs that point to image files. You can configure different cache refresh settings based on the file type.

This chapter has the following sections:

- About Templates
- Creating New Templates
- Applying Templates
- Removing Templates
- Viewing Templates
- Removing Resources

About Templates

A template is a collection of URLs, called resources. A resource might be a single URL, a group of URLs that have something in common, or an entire protocol. You name and create a template and then you assign URLs to that template by using regular expressions. This means that you can configure the proxy server to handle requests for various URLs differently. Any URL pattern you can create with regular expressions can be included in a template. Table 16-1 lists the default resources and provides some ideas for other templates.

Table 16-1 Resource regular expression wildcard patterns

Regular expression pattern	What it configures
`ftp://.*`	All FTP requests
`http://.*`	All HTTP requests
`https://.*`	All secure HTTP requests
`gopher://.*`	All Gopher requests
`connect://.*:443`	All SSL (secure) transactions to HTTPS port.
`http://home\.example\.com.*`	All documents on the `home.example.com` web site.
`.*\.gif.*`	Any URL that includes the string `.gif`
`.*\.edu.*`	Any URL that includes the string `.edu`
`http://.*\.edu.*`	Any URL going to a computer in the `.edu` domain

Understanding Regular Expressions

Proxy Server allows you to use regular expressions to identify resources. Regular expressions specify a pattern of character strings. In the proxy server, regular expressions are used to find matching patterns in URLs.

Here is an example of a regular expression:

```
[a-z]*://[^:/]*\.abc\.com.*>
```

This regular expression would match any documents from the `.abc.com` domain. The documents could be of any protocol and could have any file extension.

Table 16-2 contains regular expressions and their corresponding meanings.

Table 16-2 Regular expressions and their meanings

Expression	Meaning
.	Matches any single character except a newline.

Table 16-2 Regular expressions and their meanings

Expression	Meaning
x?	Matches zero or one occurrences of regular expression *x.*
*x**	Matches zero or more occurrences of regular expression *x.*
x+	Matches one or more occurrences of regular expression *x*.
x{n,m}	Matches the character *x* where *x* occurs at least *n* times but no more than *m* times.
x{n,}	Matches the character *x* where *x* occurs at least *n* times.
x{n}	Matches the character *x* where *x* occurs exactly *n* times.
[abc]	Matches any of the characters enclosed in the brackets.
[^abc]	Matches any character not enclosed in the brackets.
[a-z]	Matches any characters within the range in the brackets.
x	Matches the character *x* where *x* is not a special character.
\x	Removes the meaning of special character *x*.
"x"	Removes the meaning of special character *x.*
xy	Matches the occurrence of regular expression *x* followed by the occurrence of regular expression *y.*
x|y	Matches either the regular expression *x* or the regular expression *y.*
^	Matches the beginning of a string.
$	Matches the end of a string.
(x)	Groups regular expressions.

This example illustrates how you can use some of the regular expressions in Table 16-2.

```
[a-z]*://([^.:/]*[:/]|.*\.local\.com).*"
```

- `[a-z]*` matches a document of any protocol.
- `://` matches a (:) followed by (/ /).
- `[^.:/]*[:/]` matches any character string that does not include a (.),(:) or (/), and is followed by either a (:) or a (/). It therefore matches host names that are not fully qualified and hosts with port numbers.
- `|.*\.local\.com` does not match fully qualified domain name host names such as local.com but does match documents in the .local.com domain.

- `.*"` matches documents with any file extension.

NOTE As noted in Table 16-2, the backslash can be used to escape or remove the meaning of special characters. Characters such as the period and question mark have special meanings, and therefore, must be escaped if they are used to represent themselves. The period, in particular, is found in many URLs. So, to remove the special meaning of the period in your regular expression, you need to precede it with a backslash.

Understanding Wildcard Patterns

You can create lists of wildcard patterns that enable you to specify which URLs can be accessed from your site. Wildcards can be in the form of regular expressions or shell expressions, depending on usage. As a general rule:

- Use regular expressions for any pattern that matches destination URLs. This includes <Object ppath=...>, URL filters, and the NameTrans, PathCheck, and ObjectType functions.
- Use shell expressions for any pattern that matches incoming client or user IDs, including user names and groups for access control and the IP addresses or DNS names of incoming users (for example, <Client dns=...>).

You can specify several URLs by using regular expression wildcard patterns. Wildcards let you filter by domain name or by any URL with a given word in the URL. For example, you might want to block access to URLs that contain the string “careers.” To do this, you could specify `http://.*careers.*` as the regular expression for the template.

Creating New Templates

You can create a template using a regular expression wildcard pattern. You can then configure aspects that affect only the URLs specified in that template. For example, you might use one type of caching configuration for `.GIF` images and another for plain `.HTML` files.

To create a template

1. Access the Server Manager, and click the Templates tab.

   Click the Create Template link. The Create Template page displays.

2. In the Template Name field, type a name for the template you are creating, and click OK.

   The name should be something you can easily remember. The Server Manager prompts you to save and apply your changes. You can save the changes after you create a regular expression for the template, as described in the remaining steps.

Applying Templates

To apply a template

1. Access the Server Manager, and click the Templates tab.
2. Click the Apply Template link. The Apply Template page displays.
3. Type a regular expression wildcard pattern that includes all of the URLs you want to include in your template in the URL Prefix Wildcard field.
4. From the Template list, select the name of the new template you just added.
5. Click OK.
6. Click Restart Required. The Apply Changes page displays.
7. Click the Restart Proxy Server button to apply the changes.

Removing Templates

You can remove existing templates. Removing a template deletes all of the associated configurations for the template. For example, if you have access control set up for all URLs in the template TEST, removing the TEST template also removes the access control to the URLs contained in then template.

To remove a template

1. Access the Server Manager, and click the Templates tab.
2. Click the Remove Template link. The Remove Template page displays.
3. Choose the template from the Remove list.
4. Click OK.
5. Click Restart Required. The Apply Changes page displays.

6. Click the Restart Proxy Server button to apply the changes.

Viewing Templates

You can view and edit the templates created in the Server Manager.

To edit a template

1. Access the Server Manager, and click the Templates tab.
2. Click the View Template link. The View Template page displays. The templates are shown in a table that lists the regular expression for the template and the template name.
3. To edit an existing template, click the Edit Template Assignment link, which takes you to the Apply Template page.

Removing Resources

You can delete an entire regular expression object and its corresponding configurations with the Remove Resource page. For instance, you can remove the gopher resource so that all settings associated with that resource will be removed from the proxy server's configuration files.

To remove a resource

1. Access the Server Manager, and click the Templates tab.
2. Click the Remove Resource link. The Remove Resource page displays.
3. Select the resource you want to remove by choosing it from the Remove drop-down list.
4. Click OK.
5. Click Restart Required. The Apply Changes page displays.
6. Click the Restart Proxy Server button to apply the changes.

Chapter 17

Using the Client Autoconfiguration File

If you have multiple proxy servers that support many clients, you can use a client autoconfiguration file to configure all of your browser clients. The autoconfiguration file contains a JavaScript function that determines which proxy, if any, the browser uses when accessing various URLs.

When the browser starts, it loads the autoconfiguration file. Each time the user clicks a link or types in a URL, the browser uses the configuration file to determine if it should use a proxy and, if so, which proxy it should use. This feature lets you provide an easy way to configure all instances of the browser in your organization. There are several ways you can get the autoconfiguration file to your clients.

- You can use the proxy server as a web server that returns the autoconfiguration file. You point the browser to the proxy's URL. Having the proxy act as a web server lets you keep the autoconfiguration file in one place so that when you need to make updates, you need to change only one file.
- You can store the file on a web server, an FTP server, or any network directory to which the browser has access. You configure the browser to find the file by giving it the URL to the file, so any general URL will do. If you need to do complex calculations (for example, if you have large proxy chains in your organization), you might write a web server CGI program that outputs a different file depending on who accesses the file.
- You can store the autoconfiguration file locally with each copy of the browser; however, if you need to update the file, you will have to distribute copies of the file to each client.

You can create the autoconfiguration file two ways: you can use a page in the Server Manager, or you can create the file manually. Directions for creating the files appear later in this chapter.

This chapter contains the following sections:

- Understanding Autoconfiguration Files
- Using Server Manager Pages to Create Autoconfiguration Files
- Creating Autoconfiguration Files Manually

Understanding Autoconfiguration Files

This feature is documented in this book because it is likely that you, as the person administering the Proxy Server, will also create and distribute the client autoconfiguration files.

What the Autoconfiguration File Does

The autoconfiguration file is written in JavaScript, a compact, object-based scripting language for developing client and server Internet applications. The browser interprets the JavaScript file.

When the browser is first loaded, it downloads the autoconfiguration file. The file can be kept anywhere that the browser can get to it by using a URL. For example, the file can be kept on a web server. The file could even be kept on a network file system, provided the browser can get to it using a file`:// URL`.

The proxy configuration file is written in JavaScript. The JavaScript file defines a single function (called **FindProxyForURL**) that determines which proxy server, if any, the browser should use for each URL. The browser sends the JavaScript function two parameters: the host name of the system from which the browser is running and the URL it is trying to obtain. The JavaScript function returns a value to the browser that tells it how to proceed.

The autoconfiguration file makes it possible to specify different proxies (or no proxy at all) for various types of URLs, various servers, or even various times of the day. In other words, you can have multiple specialized proxies so that, for example, one serves the `.com` domain, another the `.edu` domain, and yet another serves everything else. This lets you divide the load and get more efficient use of your proxies’ disks because there is only a single copy of any file in the cache (instead of multiple proxies all storing the same documents).

Autoconfiguration files also support proxy failover, so if a proxy server is unavailable, the browser will transparently switch to another proxy server.

Accessing the Proxy as a Web Server

You can store one or more autoconfiguration files on the proxy server and have the proxy server act as a web server whose only documents are autoconfiguration files. This lets you, the proxy administrator, maintain the proxy autoconfiguration files needed by the clients in your organization. It also lets you keep the files in a central location, so if you have to update the files, you do it once and all browser clients automatically get the updates.

You keep the proxy autoconfiguration files in the *server- root* / proxy-server*id* / pac / directory. In the browser, you enter the URL to the proxy autoconfiguration file typing the URL to the file in the Proxies tab. The URL for the proxy has this format:

`http://`*proxy.domain:port/URI*

For example, the URL could be `http://proxy.example.com`. You do not need to specify a URI (part of the URL following the host:port combination); however, if you do use a URI, you can then use a template to control access to the various autoconfiguration files. For example, if you create a URI called /test that contains an autoconfiguration file called `/proxy.pac`, you can create a template with the resource pattern `http://proxy.mysite.com:8080/test/.*`. You can then use that template to set up access control specifically to that directory.

You can create multiple autoconfiguration files and have them accessed through different URLs. Table 17-1 lists some example URIs and the URLs the clients would use to access them.

Table 17-1 Sample URIs and corresponding URLs

URI (path)	URL to the proxy
/	http://proxy.mysite.com
/employees	http://proxy.mysite.com/employees
/group1	http://proxy.mysite.com/group1
/managers	http://proxy.mysite.com/managers

Using Pac Files with a Reverse Proxy

Because of the way a reverse proxy works, it can be very difficult to have a proxy server work as a reverse proxy and server `.pac` files. This is because the proxy server gets a request for a file and it needs to determine if the request is for a local `.pac` file or if the request is for a remote document.

In order to have the proxy server act as a reverse proxy in addition to maintaining and serving a .pac file, you need to manually edit the obj.conf file to make sure the order of the NameTrans functions is correct.

Create a regular mapping to have the proxy server act as a reverse proxy. This typically tells the proxy to route all requests to the remote content server. You can add a proxy autoconfiguration file and map it to a specific directory, such as /pac. In this case, any client who wants to get the .pac file would use a URL such as:

`http://proxy.mysite.com/pac`

CAUTION With this mapping, however, you must be sure that the remote content server does not have a similar directory.

Edit the obj.conf file to make sure that the directive and function for the proxy autoconfiguration file appear before any other mappings. This directive and function must be first because the proxy server normally runs through all NameTrans functions before servicing the request. However, with autoconfiguration files, the proxy immediately recognizes the path and returns the .pac file.

Here is an example from an obj.conf file that uses a reverse proxy and maintains an autoconfiguration file:

```
<Object name="default">
NameTrans from="file:" fn="map" to="ftp:"
NameTrans from="/pac" fn="pac-map" name="file" to="/ns-home/proxy/pac/proxy.pac"
NameTrans fn="redirect" from="http://foo.*" url="http://www.acme.com"
NameTrans from="/ns-icons" fn="pfx2dir" dir="/ns-home/ns-icons" name="file"
NameTrans fn="reverse-map" from="http://web.acme.com" to="http://proxy.acme.com:8080"
NameTrans fn="map" from="http://proxy.acme.com:8080" to="http://web.acme.com"
NameTrans fn="map" from="/" to="http://web.acme.com"
PathCheck fn="url-check"
Service fn="deny-service"
AddLog fn="flex-log" name="access"
AddLog fn="urldb-record"
</Object>
```

Using Server Manager Pages to Create Autoconfiguration Files

To create an autoconfiguration file using the Server Manager pages

1. Access the Server Manager, and select the Routing tab.

2. Click the Create / Edit Autoconfiguration File link. The page that appears lists any autoconfiguration files you have on your proxy's system. You can click the autoconfiguration file to edit it. The remaining steps tell you how to create a new file.

3. Type an optional URI which is the path portion of a URL, that clients will use when getting the autoconfiguration file from the proxy. For example, type / to let clients access the file as the proxy's main document (similar to an `index.html` file for a web server); clients would then use only the domain name when accessing the proxy for the autoconfiguration file. You can use multiple URIs and create separate autoconfiguration files for each URI.

4. Type a name for the autoconfiguration file using the `.pac` extension. If you have one file, you might call it simply `proxy.pac` (pac is short for proxy autoconfiguration). All autoconfiguration files are ASCII text files with a single JavaScript function.

5. Click OK. Another page appears. Use this page to create an autoconfiguration file. The items on the page are followed in order by the client. These are the items on the page:

- **Never Go Direct To Remote Server** tells Navigator to always use your proxy. You can specify a second proxy server to use in case your proxy server is not running.

- **Go Direct To Remote Server When** lets you bypass the proxy server on certain occasions. Navigator determines those occasions in the order the options are listed on the page:

    - **Connecting To Non-fully Qualified Host Names** tells Navigator to go to a server directly when the user specifies only the computer name. For example, if there is an internal web server called `winternal.mysite.com`, the user might type only `http://winternal` instead of the fully qualified domain name. In this case, Navigator goes directly to the web server instead of to the proxy.

    - **Connecting To A Host In Domain** lets you specify up to three domain names that Navigator can access directly. When specifying the domains, begin with the dot character. For example, you could type `.example.com`.

- **Connecting To A Resolvable Host** makes Navigator go directly to the server when the client can resolve the host. This option is typically used when DNS is set to resolve only local (internal) hosts. The clients would use a proxy server when connecting to servers outside of the local network.

CAUTION The above option causes the client to consult DNS for every request. It therefore, negatively impacts the performance witnessed by the client. Because of the performance impact, you should avoid using this option.

- **Connecting To A Host In Subnet** makes Navigator go directly to the server when the client accesses a server in a particular subnet. This option is useful when an organization has many subnets in a geographical area. For example, some companies might have one domain name that applies to subnets around the world, but each subnet is specific to a particular region.

CAUTION The above option causes the client to consult DNS for every request. It therefore, negatively impacts the performance witnessed by the client. Because of the performance impact, you should avoid using this option.

- **Except When Connecting To Hosts** lets you specify exceptions to the rule of going directly to a server. For example, if you type `.example.com` as a domain to which to go directly, you could make an exception for going to `home.example.com`. This tells Navigator to use your proxy when going to `home.example.com` but go directly to any other server in the `example.com` domain.
- **Secondary Failover Proxy** specifies a second proxy to use if your proxy server is not running.
- **Failover Direct** tells Navigator to go directly to the servers if your proxy server isn't running. If you specify a secondary failover proxy, Navigator tries the second proxy server before going directly to the server.

6. Click OK to create the autoconfiguration file. The file is stored in the directory *server-root*/`proxy-server`*id*/`pac`. You will get a confirmation message saying the file was created correctly. Repeat the preceding steps to create as many autoconfiguration files as you need.

Once you create your autoconfiguration file, make sure you either tell all the people using your proxy server to point to the correct autoconfiguration file or configure the copies of Navigator yourself.

Creating Autoconfiguration Files Manually

This section describes how you can manually create autoconfiguration files.

The proxy autoconfiguration file is written using client-side JavaScript. Each file contains a single JavaScript function called **FindProxyForURL** that determines which proxy server, if any, the browser should use for each URL. The browser sends the JavaScript function two parameters: the host name of the destination origin server and the URL it is trying to obtain. The JavaScript function returns a value to Navigator that tells it how to proceed. The following section describes the function syntax and the possible return values.

The FindProxyForURL Function

The syntax of the **FindProxyFor URL** function is:

```
function FindProxyForURL(url, host)
{
   ...
}
```

For every URL the browser accesses, it sends the **url** and **host** parameters and calls the function in the following way:

```
ret = FindProxyForURL(url, host);
```

url is the full URL being accessed in the browser.

host is the host name extracted from the URL that is being accessed. This is only for convenience; it is the same string as between **://** and the first **:** or **/** after that. The port number is not included in this parameter. It can be extracted from the URL when necessary.

ret (the return value) is a string describing the configuration.

The Function Return Values

The autoconfiguration file contains the function **FindProxyForURL**. As parameters, this function uses the client host name and the URL it is accessing. The function returns a single string that tells the browser how to proceed. If the string is null, no proxies should be used. The string can contain any number of the building blocks shown in Table 17-2, separated by semicolons.

Table 17-2 FindProxyForURL Return Values

Return values	Resulting action of the browser
`DIRECT`	Make connections directly to the server without going through any proxies.
`PROXY` *host:port*	Use the specified proxy and port number. If multiple values are separated by semicolons, the first proxy is used. If that proxy fails, then the next proxy is used, and so on.
`SOCKS` *host:port*	Use the specified SOCKS server. If there are multiple values separated by semicolons, the first proxy is used. If that proxy fails, then the next proxy is used, and so on.

If the browser encounters an unavailable proxy server, the browser will automatically retry the previously unresponsive proxy after 30 minutes, then after one hour, and so on, at 30-minute intervals. This means that if you temporarily shut down a proxy server, your clients will resume using the proxy no later than 30 minutes after it was restarted.

If all of the proxies are down and the DIRECT return value is not specified, the browser will ask the user if it should temporarily ignore proxies and attempt direct connections instead. Navigator will ask if proxies should be retried after 20 minutes, then again in another 20 minutes, and so on at 20-minute intervals.

In the following example, the return value tells the browser to use the proxy called `w3proxy.example.com` on port 8080, but if that proxy is unavailable, the browser uses the proxy called `proxy1.example.com` on port 8080:

```
PROXY w3proxy.example.com:8080; PROXY proxy1.example.com:8080
```

In the next example, the primary proxy is `w3proxy.example.com:8080`; if that proxy is unavailable, the browser uses `proxy1.example.com:8080.` If both proxies are unavailable, then the browser goes directly to the server (and after 20 minutes, the browser asks the user if it should retry the first proxy):

```
PROXY w3proxy.example.com:8080; PROXY proxy1.example.com:8080; DIRECT
```

JavaScript Functions and Environment

JavaScript has several predefined functions and environmental conditions that are useful with proxying. Each of these functions checks whether or not a certain condition is met and then returns a value of true or false. The related utility functions are an exception because they return a DNS host name or IP address. You can use these functions in the main **FindProxyForURL** function to determine what return value to send to the browser. See the examples later in this chapter for ideas on using these functions.

Each of the functions or environmental conditions is described in this section. The functions and environmental conditions that apply to the browser integration with the proxy are:

Hostname-based Functions

- dnsDomainIs()
- isInNet()
- isPlainhost name()
- isResolvable()
- localHostOrDomainIs()
- Related utility functions:
- dnsDomainLevels()
- dnsResolve()
- myIpAddress()
- URL / host name-based condition:
- shExpMatch()
- Time-based conditions:
- dateRange()
- timeRange()

weekdayRange()

Hostname-based Functions

The hostname-based functions let you use the host name or IP address to determine which proxy, if any, to use.

dnsDomainIs(host, domain)

The **dnsDomainIs()** function detects whether the URL host name belongs to a given DNS domain. This function is useful when you are configuring the browser not to use proxies for the local domain as illustrated in "Example 1: Proxy All Servers Except Local Hosts" on page 370 and "Example 2: Proxy Local Servers Outside the Firewall" on page 370.

This function is also useful when you are using multiple proxies for load balancing in situations where the proxy that receives the request is selected from a group of proxies based on which DNS domain the URL belongs to. For example, if you are load balancing by directing URLs containing `.edu` to one proxy and those containing `.com` to another proxy, you can check the URL host name using **dnsDomainIs()**.

Parameters:

host is the host name from the URL.

domain is the domain name to test the host name against.

Returns:

true or false

Examples:

The following statement would be true:

```
dnsDomainIs("www.example.com", ".example.com")
```

The following statements would be false:

```
dnsDomainIs("www", ".example.com")
dnsDomainIs("www.mcom.com", ".example.com")
```

isInNet(host, pattern, mask)

The **isInNet()** function enables you to resolve a URL host name to an IP address and test if it belongs to the subnet specified by the mask. This is the same type of IP address pattern matching that SOCKS uses. See "Example 4: Connect Directly to a Subnet" on page 371.

Parameters:

host is a DNS host name or IP address. If a host name is passed, this function will resolve it into an IP address.

pattern is an IP address pattern in the dot-separated format

mask is the IP address pattern mask that determines which parts of the IP address should be matched against. A value of 0 means ignore; 255 means match. This function is true if the IP address of the host matches the specified IP address pattern.

Returns:

true or false

Examples:

This statement is true only if the IP address of the host matches exactly 198.95.249.79:

```
isInNet(host, "198.95.249.79", "255.255.255.255")
```

This statement is true only if the IP address of the host matches 198.95.*.*:

```
isInNet(host, "198.95.0.0", "255.255.0.0")
```

isPlainhost name(host)

The **isPlainhost name()** function detects whether the host name in the requested URL is a plain host name or a fully qualified domain name. This function is useful if you want Netscape Navigator to connect directly to local servers as illustrated in "Example 1: Proxy All Servers Except Local Hosts" on page 370 and "Example 2: Proxy Local Servers Outside the Firewall" on page 370.

Parameters:

host is the host name from the URL (excluding port number) only if the host name has no domain name (no dotted segments).

Returns:

true if **host** is local; false if **host** is remote

Example:

```
isPlainhost name("host")
```

If **host** is something like www, then it returns true; if host is something like `www.example.com`, it returns false.

isResolvable(host)

If the DNS inside the firewall recognizes only internal hosts, you can use the **isResolvable()** function to test whether a host name is internal or external to the network. Using this function, you can configure the browser to use direct connections to internal servers and to use the proxy only for external servers. This is useful at sites where the internal hosts inside the firewall are able to resolve the DNS domain name of other internal hosts, but all external hosts are unresolvable. The **isResolvable()** function consults DNS, attempting to resolve the host name into an IP address. See "Example 3: Proxy Only Unresolved Hosts" on page 371

Parameters:

host is the host name from the URL. This tries to resolve the host name and returns true if it succeeds.

Returns:

true if it can resolve the host name, false if it cannot

Example:

```
isResolvable("host")
```

If **host** is something like `www` and can be resolved through DNS, then this function returns true.

localHostOrDomainIs(host, hostdom)

The **localHostOrDomainIs()** function specifies local hosts that might be accessed by either the fully qualified domain name or the plain host name. See "Example 2: Proxy Local Servers Outside the Firewall" on page 370.

The **localHostOrDomainIs()** function returns `true` if the host name matches the specified host name exactly or if there is no domain name part in the host name that the unqualified host name matches.

Parameters:

host is the host name from the URL.

hostdom is the fully qualified host name to match.

Returns:

true or false

Examples:

The following statement is true (exact match):

```
localHostOrDomainIs("www.example.com", "www.example.com")
```

The following statement is true (host name match, domain name not specified):

```
localHostOrDomainIs("www", "www.example.com")
```

The following statement is false (domain name mismatch):

```
localHostOrDomainIs("www.mcom.com", "www.example.com")
```

The following statement is false (host name mismatch):

```
localHostOrDomainIs("home.example.com", "www.example.com")
```

Related Utility Functions

The related utility functions enable you to find out domain levels, the host on which Netscape Navigator is running, or the IP address of a host.

dnsDomainLevels(host)

The **dnsDomainLevels()** function finds the number of DNS levels (number of dots) in the URL host name.

Parameters:
host is the host name from the URL.

Returns:
number (integer) of DNS domain levels.

Examples:

```
dnsDomainLevels("www")
```

returns 0.

```
dnsDomainLevels("www.example.com")
```

returns 2.

dnsResolve(host)

The **dnsResolve()** function resolves the IP address of the given host (typically from the URL). This is useful if the JavaScript function has to do more advanced pattern matching than can be done with the existing functions.

Parameters:
host is the host name to resolve. Resolves the given DNS host name into an IP address, and returns it in the dot-separated format as a string.

Returns:
dotted quad IP address as a string value

Example:
The following example would return the string 198.95.249.79.

```
dnsResolve("home.example.com")
```

myIpAddress()

The **myIpAddress()** function is useful when the JavaScript function has to behave differently depending on what host on which the browser is running. This function returns the IP address of the computer that is running the browser.

Returns:

dotted quad IP address as a string value

Example:

The following example returns the string 198.95.249.79 if you are running Navigator on the computer `home.example.com`.

```
myIpAddress()
```

URL/host-name-based Condition

You can match host names or URLs for load balancing and routing.

shExpMatch(str, shexp)

The **shExpMatch()** function matches either the URL host names or the URLs themselves. The main use of this function is for load balancing and intelligent routing of URLs to different proxy servers.

Parameters:

str is any string to compare (for example, the URL or the host name).

shexp is a shell expression against which to compare.

This expression is true if the string matches the specified shell expression. See "Example 6: Balance Proxy Load with shExpMatch()" on page 373.

Returns:

true or false

Examples:

The first example returns true; the second returns false.

```
shExpMatch("http://home.example.com/people/index.html",
            ".*/people/.*")
```

```
shExpMatch("http://home.example.com/people/yourpage/index.html",
            ".*/mypage/.*")
```

Time-based Conditions

You can make the **FindProxyForURL** function behave differently depending on the date, time, or day of the week.

dateRange (day, month, year...)

The **dateRange()** function detects a particular date or a range of dates, such as April 19th, 1996 through May 3rd, 1996. This is useful if you want the **FindProxyForURL** function to act differently depending on what day it is, such as if maintenance down time is regularly scheduled for one of the proxies.

The date range can be specified several ways:

```
dateRange(day)
dateRange(day1, day2)
dateRange(mon)
dateRange(month1, month2)
dateRange(year)
dateRange(year1, year2)
dateRange(day1, month1, day2, month2)
dateRange(month1, year1, month2, year2)
dateRange(day1, month1, year1, day2, month2, year2)
dateRange(day1, month1, year1, day2, month2, year2, gmt)
```

Parameters:

day is an integer between 1 and 31 for the day of month.

month is one of the month strings:
`JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC`

year is a four-digit integer for the year number (for example, 1996).

gmt is either the string GMT, which makes time comparisons occur in Greenwich Mean Time, or is left blank so that times are assumed to be in the local time zone. The GMT parameter can be specified in any of the call profiles, always as the last parameter. If only a single value is specified (from each category: day, month, year), the function returns a true value only on days that match that specification. If two values are specified, the result is true from the first time specified through the second time specified.

Examples:

This statement is true on the first day of each month, local time zone.
`dateRange(1)`

This statement is true on the first day of each month, Greenwich Mean Time.
`dateRange(1, "GMT")`

This statement is true for the first half of each month.

```
dateRange(1, 15)
```

This statement is true on the 24th of December each year.

```
dateRange(24, "DEC")
```

This statement is true on the 24th of December, 1995.

```
dateRange(24, "DEC", 1995)
```

This statement is true during the first quarter of the year.

```
dateRange("JAN", "MAR")
```

This statement is true from June 1st through August 15th, each year.

```
dateRange(1, "JUN", 15, "AUG")
```

This statement is true from June 1st, 1995, until August 15th, 1995.

```
dateRange(1, "JUN", 15, 1995, "AUG", 1995)
```

This statement is true from October 1995 through March 1996.

```
dateRange("OCT", 1995, "MAR", 1996)
```

This statement is true during the entire year of 1995.

```
dateRange(1995)
```

This statement is true from the beginning of 1995 until the end of 1997.

```
dateRange(1995, 1997)
```

timeRange (hour, minute, second...)

The **timeRange** function detects a particular time of day or a range of time, such as 9 p.m. through 12 a.m. This is useful if you want the **FindProxyForURL** function to act differently depending on what time it is.

```
timeRange(hour)
timeRange(hour1, hour2)
timeRange(hour1, min1, hour2, min2)
timeRange(hour1, min1, sec1, hour2, min2, sec2)
```

Parameters:

hour is the hour from 0 to 23. (0 is midnight, 23 is 11:00 p.m.)

min is the number of minutes from 0 to 59.

sec is the number of seconds from 0 to 59.

gmt is either the string GMT for GMT time zone, or not specified for the local time zone. This parameter can be used with each of the parameter profiles and is always the last parameter.

Returns:

true or false

Examples:

This statement is true from noon to 1:00 p.m.

```
timerange(12, 13)
```

This statement is true noon to 12:59 p.m. GMT.

```
timerange(12, "GMT")
```

This statement is true from 9:00 a.m. to 5:00 p.m.

```
timerange(9, 17)
```

true between midnight and 30 seconds past midnight.

```
timerange(0, 0, 0, 0, 0, 30)
```

weekdayRange(wd1, wd2, gmt)

The **weekdayRange()** function detects a particular weekday or a range of weekdays, such as Monday through Friday. This is useful if you want the **FindProxyForURL** function to act differently depending on the day of the week.

Parameters:

wd1 and **wd2** are any one of these weekday strings:

```
SUN MON TUE WED THU FRI SAT
```

gmt is either GMT for Greenwich Mean Time, or is left out for local time.

Only the first parameter, wd1, is mandatory. Either wd2, gmt, or both can be left out.

If only one parameter is present, the function returns a true value on the weekday that the parameter represents. If the string GMT is specified as a second parameter, times are taken to be in GMT otherwise the times are in your local time zone.

If both wd1 and wd2 are defined, the condition is true if the current weekday is between those two weekdays. Bounds are inclusive. The order of parameters is important; “MON,” “WED” is Monday through Wednesday, but “WED,” “MON” is from Wednesday to the Monday of the next week.

Examples:

The following is true Monday through Friday (local time zone).

```
weekdayRange("MON", "FRI")
```

The following is true Monday through Friday, in Greenwich Mean Time.

```
weekdayRange("MON", "FRI", "GMT")
```

The following is true on Saturdays, local time.
`weekdayRange("SAT")`

The following is true on Saturdays, in Greenwich Mean Time.
`weekdayRange("SAT", "GMT")`

The following is true Friday through Monday (the order is important)
`weekdayRange("FRI", "MON")`

Detailed Examples

Example 1: Proxy All Servers Except Local Hosts

In this example, Netscape Navigator connects directly to all hosts that are not fully qualified and the ones that are in the local domain. Everything else goes through the proxy called `w3proxy.example.com:8080`.

NOTE If the proxy goes down, connections become direct automatically.

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".example.com") ||
        dnsDomainIs(host, ".mcom.com"))
        return "DIRECT";
    else
        return "PROXY w3proxy.example.com:8080; DIRECT";
}
```

Example 2: Proxy Local Servers Outside the Firewall

This example is like the previous one, but it uses the proxy for local servers that are outside the firewall. If there are hosts (such as the main web server) that belong to the local domain but are outside the firewall and are only reachable through the proxy server, those exceptions are handled using the **localHostOrDomainIs()** function:

```
function FindProxyForURL(url, host)
{
    if ((isPlainhost name(host) ||
    dnsDomainIs(host, ".example.com")) &&
    !localHostOrDomainIs(host, "www.example.com") &&
    !localHostOrDoaminIs(host, "merchant.example.com"))
```

```
            return "DIRECT";
        else
            return "PROXY w3proxy.example.com:8080; DIRECT";
    }
```

This example uses the proxy for everything except local hosts in the `example.com` domain. The hosts `www.example.com` and `merchant.example.com` also go through the proxy.

The order of the exceptions increases efficiency: **localHostOrDomainIs()** functions get executed only for URLs that are in the local domain, not for every URL. In particular, notice the parentheses around the *or* expression before the *and* expression.

Example 3: Proxy Only Unresolved Hosts

This example works in an environment where internal DNS is set up so that it can resolve only internal host names, and the goal is to use a proxy only for hosts that aren't resolvable:

```
    function FindProxyForURL(url, host)
    {
        if (isResolvable(host))
                return "DIRECT";
            else
                return "PROXY proxy.mydomain.com:8080";
    }
```

This example requires consulting the DNS every time, so it should be grouped with other rules so that DNS is consulted only if other rules do not yield a result:

```
    function FindProxyForURL(url, host)
    {

        if (isPlainhost name(host) ||
            dnsDomainIs(host, ".mydomain.com") ||
            isResolvable(host))
            return "DIRECT";
        else
            return "PROXY proxy.mydomain.com:8080";
    }
```

Example 4: Connect Directly to a Subnet

In this example all the hosts in a given subnet are connected to directly others go through the proxy:

```
function FindProxyForURL(url, host)
{
    if (isInNet(host, "198.95.0.0", "255.255.0.0"))
        return "DIRECT";
    else

        return "PROXY proxy.mydomain.com:8080";
}
```

You can minimize the use of DNS in this example by adding redundant rules in the beginning:

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".mydomain.com") ||
        isInNet(host, "198.95.0.0", "255.255.0.0"))
        return "DIRECT";
    else
        return "PROXY proxy.mydomain.com:8080";
}
```

Example 5: Balance Proxy Load with dnsDomainIs()

This example is more sophisticated. There are four proxy servers, with one of them acting as a hot standby for the others, so if any of the remaining three goes down, the fourth one takes over. The three remaining proxy servers share the load based on URL patterns, which makes their caching more effective (there is only one copy of any document on the three servers, as opposed to one copy on each of them). The load is distributed as shown in Table 17-3.

Table 17-3 Balance Proxy Load

Proxy	Purpose
#1	`.com` domain
#2	`.edu` domain
#3	all other domains
#4	hot stand-by

All local accesses should be direct. All proxy servers run on port 8080. You can concatenate strings by using the + operator in JavaScript.

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) || dnsDomainIs(host, ".mydomain.com"))
        return "DIRECT";

    else if (dnsDomainIs(host, ".com"))
        return "PROXY proxy1.mydomain.com:8080; " +
              "PROXY proxy4.mydomain.com:8080";

    else if (dnsDomainIs(host, ".edu"))
        return "PROXY proxy2.mydomain.com:8080; " +
              "PROXY proxy4.mydomain.com:8080";

    else
        return "PROXY proxy3.mydomain.com:8080; " +
              "PROXY proxy4.mydomain.com:8080";
}
```

Example 6: Balance Proxy Load with shExpMatch()

This example is essentially the same as example 5, but instead of using **dnsDomainIs()**, this example uses **shExpMatch()**.

```
    function FindProxyForURL(url, host)
    {

    if (isPlainhost name(host) || dnsDomainIs(host, ".mydomain.com"))
        return "DIRECT";
    else if (shExpMatch(host, "*.com"))
        return "PROXY proxy1.mydomain.com:8080; " +
              "PROXY proxy4.mydomain.com:8080";
    else if (shExpMatch(host, "*.edu"))
        return "PROXY proxy2.mydomain.com:8080; " +
              "PROXY proxy4.mydomain.com:8080";
    else
        return "PROXY proxy3.mydomain.com:8080; " +
              "PROXY proxy4.mydomain.com:8080";
    }
```

Example 7: Proxying a Specific Protocol

You can set a proxy to be for a specific protocol. Most of the standard JavaScript functionality is available for use in the **FindProxyForURL()** function. For example, to set different proxies based on the protocol, you can use the **substring()** function:

```
function FindProxyForURL(url, host)
{
    if (url.substring(0, 5) == "http:") {
        return "PROXY http-proxy.mydomain.com:8080";
    }
    else if (url.substring(0, 4) == "ftp:") {
        return "PROXY ftp-proxy.mydomain.com:8080";
    }
    else if (url.substring(0, 7) == "gopher:") {
        return "PROXY gopher-proxy.mydomain.com:8080";
    }
    else if (url.substring(0, 6) == "https:" ||
            url.substring(0, 6) == "snews:") {
        return "PROXY security-proxy.mydomain.com:8080";
    }
    else {
        return "DIRECT";
    }
}
```

You can also accomplish this using the **shExpMatch()** function; for example:

```
...
if (shExpMatch(url, "http:*")) {
    return "PROXY http-proxy.mydomain.com:8080;
}

...
```

Part 5

Appendixes

Appendix A

ACL File Syntax

Access control list (ACL) files are text files containing lists that define who can access Proxy Server resources. By default the Proxy Server uses one ACL file that contains all of the lists for access to your server. Multiple ACL files can also be created and referenced in the `obj.conf` file.

Proxy Server 4 uses a different ACL file syntax than was used in Proxy Server 3.x. This appendix describes ACL files and their syntax. For detailed information about controlling access to your Proxy Server and its resources, see Chapter 8, "Controlling Access to Your Server" on page 149. Resource templates are supported in the Proxy Server 4 release, as described in Chapter 16, "Managing Templates and Resources" on page 349.

This appendix contains the following sections:

- About ACL Files and ACL File Syntax
- Referencing ACL Files in obj.conf

About ACL Files and 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 syntax version number. For example:

```
version 3.0;
```

There can be only one version line and it can appear after any comment lines. Proxy Server uses syntax version 3.0. Comments can be included in the file by using the # symbol at the start of the comment line.

Each ACL in the file begins with a statement that defines its type. ACLs can be one of three types:

- Path ACLs specify an absolute path to the resource they affect.
- Resource ACLs specify the template they affect, such as `http://`, `https://`, `ftp://`, and so on. For more information about templates, see Chapter 16, "Managing Templates and Resources" on page 349.
- 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 a named ACL can be created from the Proxy Server user interface, you must manually reference the named ACLs with resources in the `obj.conf` file.

Path and resource ACLs can include wildcards. For more information about wildcards, see Chapter 16, "Managing Templates and Resources" on page 349.

The type line begins with the letters `acl` and then includes the type information in double quotation marks, followed by a semicolon. For example:

```
acl "default";
acl "http://*.*";
```

Each type information for all ACLs must be a unique name, even among different ACL files. 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.

This section contains the following topics:

- Authentication Statements
- Authorization Statements
- The Default ACL File

Authentication Statements

ACLs can optionally specify the authentication method the server must use when processing the ACL. There are three general methods:

- Basic (default)
- Digest
- SSL

The Basic and Digest methods require the user to enter a user name and password before accessing a resource.

The SSL method requires the user to have a client certificate. To be authenticated, encryption must be turned on for the Proxy Server, and the user's certificate issuer must be in the list of trusted CAs.

By default, the server uses the Basic method for any ACL that does not specify a method. Your server's authentication database must support Digest authentication sent by a user.

Each authenticate line must specify what attribute (users, groups, or both users and groups) the server authenticates. 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 user name begins with the word `sales`:

```
allow (all) user = "sales*";
```

If the last line is changed to `group = sales`, then the ACL would fail because the group attribute is not authenticated.

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.

Writing Authorization Statements

Use the following syntax when writing authorization statements:

```
allow|deny [absolute] (right[,right...]) attribute expression;
```

Start each line with either `allow` or `deny`. It is 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_files`, and allow a few users access to the subdirectory `/my_files/personal`, the access control on the subdirectory will not work because anyone allowed access to the `/my_files` directory will also be allowed access to the `/my_files/personal` directory. To prevent this, create a rule for the subdirectory that first denies access to anyone, and then allows access for the few users who need it.

In some cases, however, if you set the default ACL to deny access to everyone, your other ACL rules do not 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. When the server receives a request for a specific resource, it builds a list of ACLs that apply for that resource. The server first adds named ACLs listed in `check-acl` statements of its `obj.conf` file. It then appends matching path and resource ACLs. This list is processed in the same order. Unless "absolute" ACL statements are present, all statements are evaluated in order. If an "absolute allow" or "absolute deny" statement evaluates to "true," the server stops processing and accepts this result.

If more than one ACL matches, 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 statement in the file and stops looking for other resources that match.

```
version 3.0;
acl "default";
authenticate (user,group) {
  prompt="Sun Java System Web Proxy Server";
};
allow (read,execute,list,info) user = "anyone";
allow (write,delete) user = "all";

acl "http://*.*";
deny (all) user = "anyone";
allow (all) user = "joe";
```

Attribute Expressions

Attribute expressions define who is allowed or denied access based on their user name, group name, host name, or IP address. The following lines provide examples of how access might be granted to different people or computers:

- `user = "anyone"`
- `user = "smith*"`
- `group = "sales"`
- `dns = "*.mycorp.com"`
- `dns = "*.mycorp.com,*.company.com"`
- `ip = "198.*"`
- `ciphers = "rc4"`
- `ssl = "on"`

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.

Use 24-hour time to specify times, such as 0400 to specify 4:00 am or 2230 for 10:30 pm. The following example restricts access to a group of users called `guests` between 8:00 am and 4:59 pm:

```
allow (read)
    (group="guests") and
    (timeofday<0800 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 have access all day on weekends, and anytime on weekdays except 8 am-4:59 pm.

```
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

Various operators can be used in attribute expressions. Parentheses delineate the operator order of precedence. The following operators can be used with `user`, `group`, `dns`, and `ip`:

- and
- or
- not
- = (equals)
- != (not equal to)

The following operators can be used with `timeofday` and `dayofweek`:

- greater than
- < less than
- = greater than or equal to
- <= less than or equal to

The Default ACL File

After installation, the *server_root*`/httpacl/generated.proxy-`*serverid*`.acl` file provides default settings for the server. The server uses the working file `genwork.proxy-`*serverid*`.acl` until settings are created in the user interface. When editing an ACL file, you could make changes in the `genwork` file, then save and apply the changes using the Proxy Server.

General Syntax Items

Input strings can contain the following characters:

- Letters `a` through `z`
- Numbers `0` through `9`
- Period and underscore

For other characters, double quotation marks must be used around the characters.

A single statement can be placed on its own line and 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

Named ACLs or separate ACL files can be referenced in the `obj.conf` file. This is done in the `PathCheck` directive using the `check-acl` function. The line has the following syntax:

```
PathCheck fn="check-acl" acl="aclname"
```

where *aclname* is the unique name of an ACL as it appears in any ACL file.

For example, you might add the following lines to the `obj.conf` file to restrict access to a directory using the ACL named `testacl`:

```
<Object ppath="https://"
PathCheck fn="check-acl" acl="testacl"
</Object
```

In the previous example, the first line is the object that states the server resource to which you want to restrict access. The second line is the `PathCheck` directive that uses the `check-acl` function to bind the named ACL (`testacl`) to the object in which the directive appears. The `testacl` ACL can appear in any ACL file referenced in `server.xml`.

Appendix B

Tuning Server Performance

Many elements impact performance in your Proxy Server environment, including the proxy client, the Proxy Server, the origin server, and the network. This appendix describes adjustments you can make that may improve Proxy Server performance.

This appendix contains the following sections:

- General Performance Considerations
- Timeout Values
- Up-to-Date Checks
- DNS Settings
- Number of Threads
- Inbound Connection Pool
- FTP Listing Width
- Cache Architecture
- Cache Batch Update
- Garbage Collection
- Solaris Performance Tuning

CAUTION This appendix is for advanced administrators ONLY. Be very careful when tuning your server, and always back up configuration files before making any changes.

General Performance Considerations

This section describes general areas to consider when analyzing Proxy Server performance.

This section contains the following topics:

- Access Logging
- ACL Cache Tuning
- Buffer Size
- Connection Timeout
- Errors Log Level
- Security Requirements
- Solaris File System Caching

Access Logging

Disabling access logging can increase the performance of your Proxy Server. This comes at a cost, however, because you lose visibility as to who is accessing the Proxy Server and what pages they are requesting.

You can disable Proxy Server access logging by commenting out the following directives in the obj.conf file:

```
Init fn="flex-init" access="$accesslog" format.access="%Ses->client.ip%
- %Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
...
AddLog fn="flex-log" name="access"
```

ACL Cache Tuning

By default the Proxy Server caches user and group authentication results in the ACL user cache. You can control the amount of time the ACL user cache is valid with 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 for the ACLCacheLifetime is 120 seconds, which means that the Proxy Server may be out of sync with the LDAP server for as long as two minutes. Setting the value to 0 (zero) turns the cache off and forces the Proxy Server to query the LDAP server each time a user authenticates. This will have a negative impact on the performance of your Proxy Server when implementing access control. If you set a large ACLCacheLifetime value, however, you may need to restart Proxy Server every time you make changes to the LDAP entries, as this will force the Proxy Server to query the LDAP server. Set a large value only if your LDAP directory is not likely to change often.

The ACLUserCacheSize is a magnus.conf parameter that configures the maximum number of entries that can be held in the cache. The default value is 200. New entries are added to the beginning of the list, and entries at the end of this list are recycled to permit new entries when the cache reaches its maximum size.

You can also set the maximum number of group memberships that can be cached per user entry with the ACLGroupCacheSize parameter. The default value is 4. Unfortunately, non-membership of a user in a group is not cached, and will result in several LDAP directory accesses on every request.

Buffer Size

You can specify the size of the send buffer (SndBufSize) and the receiving buffer (RcvBufSize) at the server's sockets. These parameters are configurable in the magnus.conf file, and the recommended values vary between various UNIX and Linux operating systems. Refer to the operating system's documentation to properly set these parameters.

Connection Timeout

You can specify the number of seconds the server waits for data to arrive from the client before closing the connection by using the AcceptTimeout directive in the magnus.conf file. If data does not arrive before the timeout expires, the connection is closed. This is set to 30 seconds by default. Under most circumstances, you do not need to change this setting. You can free up threads by setting this to less than the default, but you might also disconnect users with slower connections.

Errors Log Level

Increasing the `loglevel` attribute in the `LOG` tag of the `server.xml` file causes the server to generate and store more information in the errors log. This does, however, come at a cost when writing entries to that file. Increase logging only while debugging a problem, and minimize logging when not in a troubleshooting mode.

Security Requirements

Enabling SSL increases the privacy and security of your Proxy Server, but also impacts performance as encryption and decryption of the packets causes overhead. You may want to consider offloading encryption and decryption processing to a hardware accelerator card.

Solaris File System Caching

The Proxy Server cache is not stored in random access memory. Accesses to files are made to the file system each time a document is extracted from cache. You may want to consider using Solaris' file system caching to pre-load the Proxy Server cache into memory. References to cached files are then extracted from memory rather than from the file system.

Timeout Values

Timeouts have a significant impact on server performance. Setting the optimal timeout for the Proxy Server helps to conserve network resources.

There are two instance-specific SAFs (server application functions) and one global parameter that can be used to configure timeout values within the Proxy Server.

This section contains the following topics:

- init-proxy SAF (obj.conf)
- http-client-config SAF (obj.conf)
- KeepAliveTimeout (magnus.conf)

init-proxy SAF (obj.conf)

The `init-proxy` function initializes the Proxy Server's internal settings. This function is called during the initialization of the Proxy Server, but should also be specified in the `obj.conf` file to ensure that the values are initialized properly.

The syntax of this function is as follows:

```
Init fn=init-proxy
  timeout=seconds
  timeout-2=seconds
```

In the previous example, the following parameters have direct applicability to Proxy Server timeout settings for the `init-proxy` SAF:

- `timeout` (proxy timeout) – The proxy timeout parameter tells the server how long to wait before aborting an idle connection. A high proxy timeout value commits a valuable proxy thread to a potentially down client for a long time. A low timeout value aborts CGI scripts that take a long time to produce results, such as a database query gateway.

  To determine the best proxy timeout for the server, consider these issues:

  - Will the Proxy Server be handling many database queries or CGI scripts?
  - Will the Proxy Server be handling a small enough number of requests that a process can be spared at any given time?

  If you answered yes to either of these questions, you may decide to set a high proxy timeout value. The highest proxy timeout value recommended is 1 hour. The default value is 300 seconds (5 minutes).

  You can view or modify the proxy timeout value by accessing the Configure System Preferences page under the Preferences tab in the Server Manager. This parameter is referenced as Proxy Timeout.

- `timeout-2` (timeout after interrupt) – The timeout after interrupt value tells the Proxy Server how much time it must continue writing a cache file after a client has aborted the transaction. In other words, if the Proxy Server has almost finished caching a document and the client aborts the connection, the server can continue caching the document until it reaches the timeout after interrupt value.

  The highest recommended timeout after interrupt value is 5 minutes. The default value is 15 seconds.

http-client-config SAF (obj.conf)

The `http-client-config` function configures the Proxy Server's HTTP client.

The syntax of this function is as follows:

```
Init fn=http-client-config
  keep-alive=(true|false)
  keep-alive-timeout=seconds
  always-use-keep-alive=(true|false)
  protocol=HTTP Protocol
  proxy-agent="Proxy-agent HTTP request header"
```

The settings are defined as follows:

- `keep-alive` – (Optional) Boolean that indicates whether the HTTP client should attempt to use persistent connections. The default is true.
- `keep-alive-timeout` – (Optional) The maximum number of seconds to keep a persistent connection open. The default is 29.
- `always-use-keep-alive` – (Optional) Boolean that indicates whether the HTTP client can reuse existing persistent connections for all types of requests. The default is false, meaning persistent connections will not be reused for non-GET requests or for requests with a body.
- `protocol` – (Optional) HTTP protocol version string. By default, the HTTP client uses either HTTP/1.0 or HTTP/1.1, based on the contents of the HTTP request. In general, do not use the protocol parameter unless you encounter specific protocol interoperability problems.
- `proxy-agent` – (Optional) Value of the Proxy-agent HTTP request header. The default is a string that contains the Proxy Server product name and version.

KeepAliveTimeout (magnus.conf)

This parameter determines the maximum time (in seconds) that the server holds open HTTP keep-alive connections or persistent connections between the client and the Proxy Server. The default is 30 seconds. The connection times out if idle for more than 30 seconds. The maximum is 300 seconds (5 minutes).

NOTE Timeout settings in the `magnus.conf` file apply to connections between the client and the Proxy Server. Timeout settings in the `http-client-config` SAF in the `obj.conf` file apply to connections between the Proxy Server and an origin server.

Up-to-Date Checks

Proxy Servers increase performance by serving documents out of a local cache rather than obtaining them from the origin server. One drawback to this methodology is the potential to provide documents that are out of date.

The Proxy Server can perform a check to determine whether a document is up to date, and then refresh the cached version if it is determined that the document is old. This up-to-date check should be performed only as necessary, as frequently checking documents can decrease the overall performance of the Proxy Server.

Up-to-date checking is configured on the Set Cache Specifics page of the Caching tab. The default is to check for a new document every two hours. This information is configured in the `ObjectType` directive with the `max-uncheck` parameter.

To improve the server's performance while ensuring that a document is up-to-date, customize up-to-date checking by determining a reasonable document lifetime in conjunction with the last-modified factor, which is discussed below.

Last-Modified Factor

The last-modified factor is used to fine-tune document up-to-date processing. This factor helps determine the likelihood that a document will change based on the previous changes that have been noted.

The last-modified factor is a fraction between .02 and 1.0. It is multiplied by the interval between a document's actual last modification and the time the last up-to-date check was performed on the document. The resulting number is compared with the time since the last up-to-date check. If the number is smaller than the time interval, the document has not expired. If, however, it is larger than the time interval, then the document has expired and a new version is obtained from the origin server.

The last-modified factor enables you to ensure that recently changed documents are checked more often than old documents.

You should set a last-modified factor between 0.1 and 0.2.

DNS Settings

DNS is the system used to associate standard IP addresses with host names. This system can tie up valuable Proxy Server resources if not configured wisely. To optimize performance, consider the following:

- Enable DNS Caching

  DNS Caching is enabled by choosing the Configure DNS Cache link under the Preferences tab of the Server Manager. Select the Enabled radio button for DNS caching.

- Do not log client DNS names – log only client IP addresses

  Client DNS name logging is disabled by choosing the Set Access Log Preferences link under the Server Status tab of the Server Manager. Select the IP Addresses radio button to log IP addresses rather than client host names.

- Disable reverse DNS

  Reverse DNS translates IP addresses into host names. Reverse DNS is disabled by choosing the Configure System Preferences link under the Preferences tab of the Server Manager. Select the No radio button to disable reverse DNS.

- Avoid access control based on client host names

  Use clients' IP addresses instead of host names (if possible) in access control statements.

Number of Threads

The `RqThrottle` parameter in the `magnus.conf` file specifies the maximum number of simultaneous transactions the Proxy Server can handle. The default value is 128. Changes to this value can be used to throttle the server, minimizing latencies for transactions that are performed.

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 the maximum number of requests is already being processed. If the limit has been reached, the processing of new requests is deferred until the number of active requests drops below the maximum amount.

You can monitor the number of simultaneous requests by viewing the `SessionCreationInfo` portion of data generated by perfdump, or `proxystats.xml` data. From this information, you can determine the maximum number of simultaneous (peak) requests as compared to the total number (limit) of threads. The following information came from a perfdump output:

```
SessionCreationInfo:
------------------------
Active Sessions         1
Keep-Alive Sessions     0
Total Sessions Created  48/128
```

Active Sessions shows the number of sessions (request processing threads) currently servicing requests. Keep-Alive Sessions is similar to Active Sessions, but is specific to whether the client is requesting a keep-alive connection. Total Sessions Created shows both the number of sessions created and the maximum number of sessions allowed. These are the minimum and maximum values for the RqThrottle value.

NOTE RqThrottleMin is the minimum number of threads the server initiates upon startup. The default value is 48. This parameter can also be set in the magnus.conf file, but does not appear by default.

Reaching the maximum number of configured threads is not necessarily undesirable, and you do not need to automatically increase the RqThrottle value if you see this. Reaching this limit means that the server needed this many threads at peak load, but as long as it was able to serve requests in a timely manner, the server is adequately tuned. However, at this point, connections will queue up in the connection queue, potentially overflowing it. If you check your perfdump output on a regular basis and notice that total sessions created is often near the RqThrottle maximum, consider increasing your thread limits.

Suitable RqThrottle values range from 100 to 500, depending on the load.

Inbound Connection Pool

The inbound connection pool can be tuned using the KeepAlive* and related settings in magnus.conf, including the following:

- MaxKeepAliveConnections
- KeepAliveThreads
- KeepAliveTimeout
- KeepAliveQueryMaxSleepTime
- KeepAliveQueryMeanTime

- `ConnQueueSize`
- `RqThrottle`
- `acceptorthreads`

NOTE For more information about these parameters, see Chapter 2 of the Sun ONE Web Server 6.1 *Performance Tuning, Sizing, and Scaling Guide* at:

`http://docs.sun.com/source/817-6249/index.html`

Outbound connection pool settings cannot be configured in this release of the Proxy Server.

FTP Listing Width

You might want to modify the width of FTP listings to better suit your needs. Increasing listing width allows longer file names and thus reduces file name truncation. The default width is 80 characters.

The FTP listing width can be modified by choosing the Tune Proxy link under the Preferences tab of the Server Manager.

Cache Architecture

Performance of your server can be improved by architecting your cache wisely. Suggestions to keep in mind when architecting your cache:

- Distribute the load
- Use multiple proxy cache partitions
- Use multiple disk drives
- Use multiple disk controllers

Proper cache setup is critical to the performance of your Proxy Server. The most important rule to remember when laying out your proxy cache is to distribute the load. Caches should be set up with approximately 1 Gbyte per partition and should be spread across multiple disks and multiple disk controllers. This type of arrangement provides faster file creation and retrieval than is possible with a single, larger cache.

Cache Batch Update

The cache batch update feature enables you to pre-load files from a specified web site or perform an up-to-date check on documents already in the cache. This is typically initiated when the load on the Proxy Server is at its lowest. From the Cache Batch Updates form you can create, edit, and delete batches of URLs and enable and disable batch updating.

You can actively cache content (as opposed to on-demand), by specifying files to be batch updated. The Proxy Server enables you to perform an up-to-date check on several files currently in the cache, or pre-load multiple files in a particular web site.

At larger sites with a network of servers and proxies, administrators might want to use batch updating to pre-load a given area of the web. The batch process would perform a recursive descent across links in the document and cache the content locally. This function can be a burden on remote servers, so be careful when using it. Measures are taken to keep the process from performing recursion indefinitely, and the parameters in the `bu.conf` configuration file provides some control of this process.

Use the Proxy Server access logs to determine which sites are the most commonly active and perform a batch update on those sites to increase performance.

Garbage Collection

Garbage collection is the process of reviewing the Proxy Server cache and removing old (stale) files. Garbage collection is a resource-intensive process, therefore you may need to tune some garbage collection settings to improve its performance.

The following parameters provide the ability to fine-tune the garbage collection process. You can view or modify these parameters on the Tune Garbage Collection form, which is located by choosing Tune GC under the Caching tab of the Server Manager.

This section contains the following topics:

The gc hi margin percent Variable

The `gc hi margin percent` variable controls the percentage of the maximum cache size that, when reached, triggers garbage collection.

This value must be higher than the value for `gc lo margin percent`.

The valid range for `gc hi margin percent` is 10 to 100 percent. The default value is 80 percent (trigger garbage collection when the cache is 80 percent full).

The gc lo margin percent Variable

The `gc lo margin percent` variable controls the percentage of the maximum cache size that the garbage collector targets.

This value must be lower than the value for `gc hi margin percent`.

The valid range for `gc lo margin percent` is 5 to 100 percent. The default value is 70 percent (target at 70 percent full cache after garbage collection).

The gc extra margin percent Variable

If the garbage collection is triggered by a reason other than the partition's size getting close to the maximum allowed size (`gc hi margin percent`), the garbage collector will use the percentage set by the `gc extra margin percent` variable to determine the fraction of the cache to remove.

The valid range for `gc extra margin percent` is 0 to 100 percent. The default value is 30 percent (remove 30 percent of existing cache files).

The gc leave fs full percent Variable

The gc leave fs full percent value determines the percentage of the cache partition size below which garbage collection will not go. This value prevents the garbage collector from removing all files from the cache if some other application is monopolizing the disk space.

The valid range for gc leave fs full percent is 0 (allow total removal) to 100 percent (remove nothing). The default value is 60 percent (allow the cache size to shrink to 60 percent of current).

Solaris Performance Tuning

Various parameters in the Solaris kernel can be used to fine-tune Proxy Server performance. The following table lists some of those parameters.

Table B-1 Solaris Performance Tuning Parameters

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, and pipes, if any).
rlim_fd_cur	/etc/system	64	8192	
sq_max_size	/etc/system	2	0	Controls streams driver queue size. Setting this to 0 makes it infinity, so the performance runs will not be hit by lack of buffer space. Set this on clients, too.
tcp_close_wait_interval	ndd/dev/tcp	240000	60000	Set this 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.

Table B-1 Solaris Performance Tuning Parameters

Parameter	Scope	Default Value	Tuned Value	Comments
tcp_rexmit_interval_initial	ndd/dev/tcp	3000	3000	If retransmission is greater than 30-40%, 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 this 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	Use this to increase the transmit buffer.
tcp_recv_hiwat	ndd/dev/tcp	8129	32768	Use this to increase the receive buffer.

For more information about these parameters, see Chapter 5 of the Sun ONE Web Server 6.1 *Performance Tuning, Sizing, and Scaling Guide* at:

http://docs.sun.com/source/817-6249/index.html

Index