Administrator's Guide

Sun™ ONE Web Server

Version 6.1

817-1831-10
August 2003

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

Contents

About This Guide

This guide describes how to configure and administer Sun™ Open Net Environment (Sun ONE) Web Server 6.1. It is intended for information technology administrators in the corporate enterprise who want to extend client-server applications to a broader audience through the World Wide Web.

This preface includes the following sections:

What's In This Guide?

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

After you install the server, this guide is available in HTML format at `/manual/https/ag` in your server root directory. By default, the server root directory is `C:\Sun\WebServer6.1\` or `/opt/SunWwbsvr`.

How This Guide Is Organized

This guide is divided into five parts, plus a glossary, and a comprehensive index. If you are new to Sun ONE Web Server 6.1, begin with Part I, "Server Basics" for an overview of the product. If you are already familiar with this version of Sun ONE Web Server, skim the material in Part I, "Server Basics" before going on to Part II, "Using the Administration Server."

Once you are familiar with the fundamentals of using the Administration Server, you can refer to Part III, "Configuring, Monitoring, and Performance Tuning," which includes examples of how to configure and monitor your Sun ONE Web Servers. Part IV, "Managing Virtual Servers and Services" provides information for using programs and configuration styles.

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

Part I: Server Basics

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

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

Part II: Using the Administration Server

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

- Chapter 5, "Setting Administration Preferences" describes how to use the Administration Server Preferences and Global Settings forms to configure your Sun ONE Web Servers.

- Chapter 3, “Managing Users and Groups” describes how to how to use the Administration Server Users and Groups forms to configure your Sun ONE Web Servers.
- Chapter 4, “J2EE-based Security for Web Container and Web Applications” describes how to configure your Sun ONE Web Server security and discusses two security models: ACL-based access control and Java™ 2 Platform, Enterprise Edition (J2EE™) / Servlets-based authentication and authorization.
- Chapter 6, “Using Certificates and Keys”describes how you can use certificates and public keys to enhance security. Note that before reading this chapter you should be familiar with the basic concepts of public-key cryptography and the Secure Sockets layer (SSL) protocol. These concepts include encryption and decryption; keys; digital certificates and signatures; and SSL encryption, ciphers, and the major steps of the SSL handshake.
- Chapter 7, “Managing Server Clusters” describes the concept of clustering servers and explains how you can use them to share configurations among servers.

Part III: Configuring and Monitoring

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

- Chapter 8, “Configuring Server Preferences” describes how to configure server preferences for your Sun ONE Web Server.
- Chapter 9, “Controlling Access to Your Server” describes how to specify who can access parts of your server.
- Chapter 10, “Using Log Files” describes how to monitor your Sun ONE Web Server using the Hypertext Transfer Protocol (HTTP), by recording and viewing log files, or by using the performance monitoring tools provided with your operating system.
- Chapter 11, “Monitoring Servers” describes how to monitor your Sun ONE Web Server using SNMP (Simple Network Management Protocol).
- Chapter 12, “Configuring Naming and Resources” describes how you can configure Java Naming and Description Interface (JNDI) resources and include database connectivity on your server.

Part IV: Managing Virtual Servers and Services

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

- Chapter 13, "Using Virtual Servers" describes how to set up and administer virtual servers using your Sun ONE Web Server.
- Chapter 14, "Creating and Configuring Virtual Servers" describes how you can create and configure individual virtual servers.
- Chapter 15, "Extending Your Server With Programs" describes how to install Java applets, CGI programs, JavaScript applications, and other plug-ins onto your server.
- Chapter 16, "Content Management" describes how you can configure and manage your server's content.
- Chapter 17, "Applying Configuration Styles" describes how to use configuration styles with Sun ONE Web Server.
- Chapter 18, "Using Search" describes how to search the contents and attributes of documents on the server. In addition, this chapter describes how to create a customized text search interface that's tailored to your user community.
- Chapter 19, "Web Publishing with WebDAV" describes how you can configure a virtual server to use the WebDAV protocol that enables web publishing and in-place collaborative web authoring.

Part V: Appendixes

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

- Appendix A, "Command Line Utilities" provides instructions for using command line utilities in place of the user interface screens.
- Appendix B, "Hypertext Transfer Protocol" provides a short introduction to a few HTTP basic concepts.
- Appendix C, "ACL File Syntax" describes the access-control list (ACL) files and their syntax.
- Appendix D, "Support for Internationalization and Localization" describes the internationalized version of the Sun ONE Web Server.

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

Using the Sun ONE Web Server Documentation

The Sun ONE Web Server manuals are available as online files in PDF and HTML formats at:

`http://docs.sun.com/prod/sunone`

The following table lists the tasks and concepts described in the Sun ONE Web Server manuals.

Table 1 Sun ONE Web Server Documentation Roadmap

For Information About	See the Following
Late-breaking information about the software and documentation	*Release Notes*
Getting started with Sun ONE Web Server, including hands-on exercises that introduce server basics and features (recommended for first-time users)	*Getting Started Guide*
Performing installation and migration tasks: • Installing Sun ONE Web Server and its various components, supported platforms, and environments • Migrating from Sun ONE Web Server 4.1 or 6.0 to Sun ONE Web Server 6.1	*Installation and Migration Guide*

Table 1 Sun ONE Web Server Documentation Roadmap

For Information About	See the Following
Performing the following administration tasks: • Using the Administration and command-line interfaces • Configuring server preferences • Using server instances • Monitoring and logging server activity • Using certificates and public key cryptography to secure the server • Configuring access control to secure the server • Using Java™ 2 Platform, Enterprise Edition (J2EE™ platform) security features • Deploying applications • Managing virtual servers • Defining server workload and sizing the system to meet performance needs • Searching the contents and attributes of server documents, and creating a text search interface • Configuring the server for content compression • Configuring the server for web publishing and content authoring using WebDAV	*Administrator's Guide*
Using programming technologies and APIs to do the following: • Extend and modify Sun ONE Web Server • Dynamically generate content in response to client requests • Modify the content of the server	*Programmer's Guide*

Table 1 Sun ONE Web Server Documentation Roadmap

For Information About	See the Following
Creating custom Netscape Server Application Programmer's Interface (NSAPI) plugins	*NSAPI Programmer's Guide*
Implementing servlets and JavaServer Pages™ (JSP™) technology in Sun ONE Web Server	*Programmer's Guide to Web Applications*
Editing configuration files	*Administrator's Configuration File Reference Guide*
Tuning Sun ONE Web Server to optimize performance	*Performance Tuning, Sizing, and Scaling Guide*

Documentation Conventions

This section describes the types of conventions used throughout this guide:

- **File and directory paths** are given in UNIX® format (with forward slashes separating directory names). For Windows versions, the directory paths are the same, except that backslashes are used to separate directories.
- **URLs** are given in the format:

  `http://`*`server.domain/path/file`*`.html`

  In these URLs, **server** is the server name where applications are run; **domain** is your Internet domain name; **path** is the server's directory structure; and **file** is an individual filename. Italic items in URLs are placeholders.
- **Font conventions** include:
  - The `monospace` font is used for sample code and code listings, API and language elements (such as function names and class names), file names, pathnames, directory names, and HTML tags.
  - *`Italic`* type is used for code variables.
  - *Italic* type is also used for book titles, emphasis, variables and placeholders, and words used in the literal sense.
  - **Bold** type is used as either a paragraph lead-in or to indicate words used in the literal sense.
- **Installation root directories** are indicated by *install_dir* in this document.

By default, the location of *install_dir* on UNIX-based platforms is:

`/opt/SUNWwbsvr/`

On Windows, it is:

`C:\Sun\WebServer6.1`

Product Support

If you have problems with your system, contact customer support using one of the following mechanisms:

- The online support web site at:

  `http://www.sun.com/supportraining/`

Part 1

Server Basics

Chapter 1

Introduction to Sun ONE Web Server

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

This chapter includes the following sections:

Sun ONE Web Server

Sun ONE Web Server 6.1 is a multi-process, multi-threaded, secure web server built on open standards. It provides high performance, reliability, scalability, and manageability for any size enterprise.

This section describes the features of Sun ONE Web Server and introduces some of the basic administration tasks you can perform. It includes the following topics:

What's New in Sun ONE Web Server 6.1

Sun ONE Web Server 6.1 includes the following new features:

Java Servlet 2.3 and JavaServer Pages (JSP) 1.2 Support

Sun ONE Web Server 6.1 includes a Java™ 2 Platform, Enterprise Edition (J2EE™)-compliant implementation of the Java™ Servlet 2.3 and JavaServer Pages™ (JSP™) 1.2 specifications. A J2EE-compliant web container provides the flexibility and reliability needed to design and deploy web applications that comply with Java™ technology standards. Web applications can be deployed on a per virtual server basis.

For information about these technologies, see the following resources:

Java Servlets
http://java.sun.com/products/servlet/index.html

Java Servlet 2.3 specification
http://java.sun.com/products/servlet/download.html

JavaServer Pages
http://java.sun.com/products/jsp/index.html

For information about developing servlets and JSPs in Sun ONE Web Server, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*.

JDK 1.4.1_03 Support

Sun ONE Web Server 6.1 supports the Java Developer's Kit (JDK™) 1.4.1_03. This JDK is bundled with the Web Server and installed during installation (if you choose to install it). You can also install your own JDK at a later time, after you install the Web Server. If you plan to use the Administration Server and Java and servlet support, a JDK must be installed.

WebDAV Support

Sun ONE Web Server 6.1 supports the Web-based Distributed Authoring and Versioning (WebDAV) protocol, which enables collaborative web publishing with the following features:

Compliance with RFC 2518 and interoperability with RFC 2518 clients

- Security and access control for web publishing
- Basic publishing operations on file-system-based WebDAV collections and resources

WebDAV provides integrated support for content metadata, name space management, and overwrite protection. These technologies, combined with the many authoring tools that support WebDAV, provide an ideal development platform for collaborative environments.

NSAPI Filters Support

Sun ONE Web Server 6.1 extends the Netscape Server Application Programmer's Interface (NSAPI) to support NSAPI filters.

Filters enable the custom processing of HTTP request and response streams, allowing a function to intercept and potentially modify the content presented to or generated by another function. For example, a plugin could install an NSAPI filter to intercept an XML page generated by another plugin's Server Application Function (SAF), then transform that XML page into an HTML, XHTML, or WAP page appropriate for the client. Alternatively, an NSAPI filter could decompress data received from a client before presenting it to another plugin.

For more information, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide*.

HTTP Compression Support

Sun ONE Web Server 6.1 supports content compression, which allows you to increase delivery speed to clients and serve higher content volumes without incurring a corresponding increase in hardware expenses. Content compression reduces content download time, a benefit most apparent to users of dial-up and high-traffic connections.

For more information, see the Sun ONE Web Server 6.1 *Administrator's Guide.*

New Search Engine Support

Sun ONE Web Server 6.1 supports a new Java-based search engine that provides full-text search indexing and retrieval. The search feature allows users to search documents on the server and display results on a web page. Server administrators create the indexes of documents against which users will search, and can customize the search interface to meet specific needs.

For more information, see the Sun ONE Web Server 6.1 *Administrator's Guide.*

Enhanced Security

New functionality in Sun ONE Web Server 6.1 allows you to restrict access using flat file authentication. Unlike previous versions of the Web Server, Sun ONE Web Server 6.1 now also supports the Java Security Manager. The Java Security Manager is disabled by default when you install the product. For more information about `server.xml`, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference Guide.*

JNDI Support

Sun ONE Web Server 6.1 supports the Java Naming and Directory Interface™ (JNDI), which provides seamless connectivity to heterogeneous enterprise naming and directory services.

JDBC Support

Sun ONE Web Server provides out-of-the-box, seamless Java™ DataBase Connectivity (JDBC™), and supports a wide range of industry-standard and customized JDBC drivers.

Sun ONE Studio 5 Support

Sun ONE Web Server 6.1 supports Sun™ ONE Studio 5, Standard Edition. Sun ONE Studio technology is Sun's powerful, extensible, integrated development environment (IDE) for Java technology developers. Sun ONE Studio 5 is based on NetBeans™ software, and integrated with the Sun ONE platform. (Sun ONE Web Server 6.1 also supports NetBeans 3.5 and 3.5.1.)

Sun ONE Studio support is available on all platforms supported by Sun ONE Web Server 6.1. The plugin for the Web Server can be obtained in the following ways:

- From the Companion CD in the Sun ONE Web Server 6.1 media kit
- By using the AutoUpdate feature of Sun ONE Studio
- From the download center for Sun ONE Web Server 6.1 at http://www.sun.com/software/download/inter_ecom.html

It is important to note that the Sun ONE Studio 5 plugin for Sun ONE Web Server 6.1 works only with a local Web Server (that is, with the IDE and the Web Server on the same machine).

The behavior of the Sun ONE Studio 5 plugin for Sun ONE Web Server 6.1 is the same as that for Sun™ ONE Application Server 7. For information about using the web application features in Sun ONE Studio 5, see the tutorial at the following location:
http://developers.sun.com/tools/javatools/documentation/s1s5/cdshop.pdf

Set the Sun ONE Web Server 6.1 instance as the default, and then take the same actions described in the tutorial.

Also see the following NetBeans tutorial
http://usersguide.netbeans.org/tutorials/webapps/index.html

For more information about Sun ONE Studio 5, visit
http://www.sun.com/software/sundev/jde/

NSS 3.3.5 and NSPR 4.1.5 Support

Sun ONE Web Server 6.1 supports Network Security Services (NSS) 3.3.5 and Netscape Portable Runtime (NSPR) 4.1.5.

PHP Compatibility

Sun ONE Web Server 6.1 is compatible with PHP, the versatile and widely-used Open Source web scripting language. PHP (a recursive acronym for PHP: Hypertext Preprocessor) runs on all major operating systems.

PHP version 4.3.2 is recommended for use with Sun ONE Web Server 6.1. For PHP-related installation and configuration information specific to Sun ONE Web Server, see
http://www.php.net/manual/en/install.netscape-enterprise.php

Enhanced Hardware Accelerator Encryption Support

Sun ONE Web Server 6.1 provides hardware accelerator support for Sun™ Crypto Accelerator 1000, a cryptographic accelerator board that enhances the performance of SSL on the Web Server.

Start on Boot Option

On UNIX platforms, Sun ONE Web Server 6.1 introduces the Start on Boot option, which allows you to configure the Web Server to be started automatically when the system boots. For more information, see the Sun ONE Web Server 6.1 *Installation and Migration Guide.*

Additional Features

Support for multiple processes and process monitors, failover, automatic recovery, and dynamic log rotation.

Administering and Managing Sun ONE Web Servers

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

- Sun ONE Web Server Administration Server
- Server Manager
- Class Manager
- Virtual Server Manager

In previous releases, the Web Server and other Netscape servers were administered by a single server, called the Administration Server. In the 4.x release, the “administration server” became simply an additional instance of the Sun ONE Web Server, called Sun ONE Web Server Administration Server, or Administration Server. You use the Administration Server to administer all of your Sun ONE Web Server instances. For more information, see “Administration Server” on page 37..

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

For managing individual instances of Sun ONE Web Server, you can use the Server Manager. For more information, see “Server Manager” on page 38.

To manage virtual servers, use the Class Manager. For more information, see “Class Manager” on page 39.

Sun ONE Web Server Configuration

Sun ONE Web Server is configured to enable you to turn on or off various features, determine how to respond to individual client requests, and write programs that run on and interact with the server’s operation. The instructions (called directives) which identify these options are stored in configuration files. Sun ONE Web Server reads the configuration files on startup and during client requests to map your choices with the desired server activity.

For more information about these files, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference.*

Administration Server

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

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

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

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

Before you can get to any forms, the Administration Server prompts you to authenticate yourself. This means you need to type a user name and password. You set up the "superuser" user name and password when you install Sun ONE Web Server on your computer. The following figure shows a typical authentication screen:

After installation, you can use distributed administration to give multiple people access to different forms in the Administration Server. For more information about distributed administration, see "Allowing Multiple Administrators" on page 102 in Chapter 5, "Setting Administration Preferences".

The settings for the Administration Server appear in the right pane, organized by a set of tabs.

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

- Servers
- Preferences
- Global Settings
- Users and Groups

- Security
- Cluster Mgmt (Cluster Management)

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

For more information on using the Administration Server, including information regarding these administration-level tasks, see "Administering Sun ONE Web Servers" on page 45

Server Manager

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

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

1. Install and start your Sun ONE Web Server.

   The Administration Server displays the Servers page.

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

   Sun ONE Web Server displays the Server Manager Preferences page.

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

You use the links on the Preferences page to manage options such as thread pool settings, and to turn the web server on and off.

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

- Security
- Logs
- Monitor
- Virtual Server Class

- Java

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

Class Manager

The Class Manager is a web-based interface that contains the Java forms you use to configure your virtual Sun ONE Web Servers. The user interface for virtual servers has two parts, the Server Manager and the Class Manager. The Class Manager contains settings that affect a single class or single virtual server. You can set services for the class in the Class Manager, as well as add virtual servers (members of the class) and configure settings for an individual virtual server.

You can access the Class Manager for Sun ONE Web Server by performing the following steps:

1. From the Server Manager, click the Virtual Server Class tab.

   The Server Manager displays the Manage a Class of Virtual Server page.

2. From the drop-down list, select a virtual server class and click Manage.

   Sun ONE Web Server displays the Class Manager's Select a Virtual Server page.

   You can also access the Class Manager by simply clicking the Class Manager link in the upper right-hand corner of the screen.

The Class Manager provides the following tabs to manage your Sun ONE Web Server virtual servers:

- Virtual Servers
- Programs
- Content Management
- Styles

For more information, see the Class Manager in the online help.

Virtual Server Manager

To access the Virtual Server Manager, go to the Virtual Servers tab in the Class Manager, then select a virtual server from the list on the Manager Virtual Servers page and click Manage, or click on the link to a virtual server under the tree view.

The pages provided in the Virtual Server Manager allow you to check the status and settings, set the Java web applications state to on, and generate reports for the selected virtual server.

The Virtual Server Manager provides the following tabs to manage your Sun ONE Web Server virtual servers:

- Preferences
- Logs
- Web Applications
- WebDAV
- Search

Using the Resource Picker

Most of the Server Manager and Class Manager pages configure the entire Sun ONE Web Server or an entire class. However, some pages can configure either the entire server (or class) or files and directories that the server (or class) maintains. These pages have the Resource Picker at the top of the page.

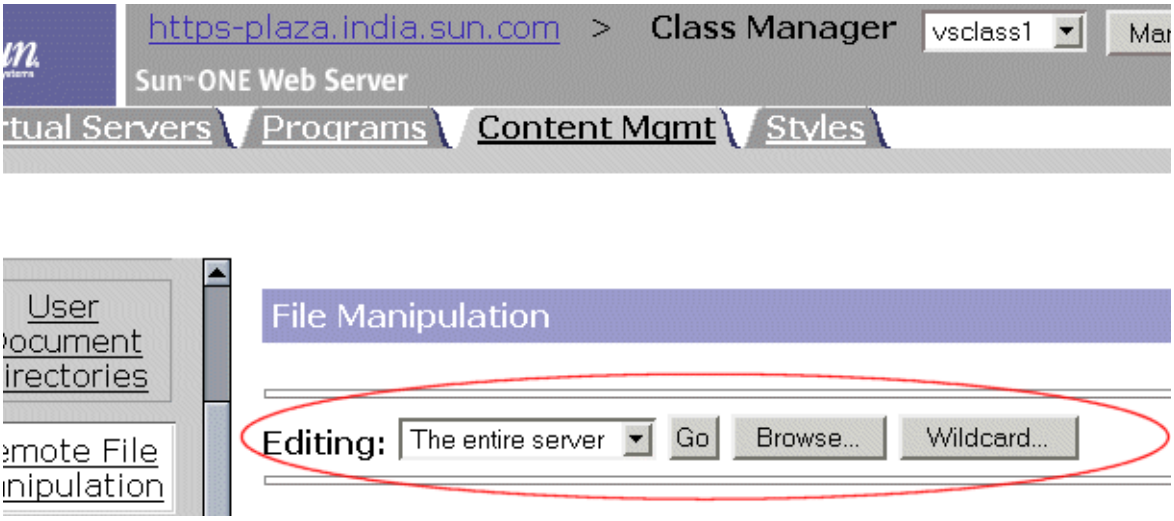


Figure 1-1 The Resource Picker

The Resource Picker appears on a number of pages, including the Server Manager's Log Preferences page and most screens accessible from the Class Manager's Content Management tab.

To use the Resource Picker, choose a resource from the drop-down list for configuration. Click Browse to browse your primary document directly; click Wildcard to configure files with a specific extension.

Wildcards Used in the Resource Picker

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

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

Wildcard patterns are applied on the directory path and not just on the filename. A wildcard pattern is therefore applicable to files in a particular directory only. For example, to add files to a directory `/tmp`, you could specify the wildcard pattern `tmp/*.html`. To add `index.html` from all subdirectories, the pattern would be `*/index.html`.

Table 1 Resource Picker wildcard patterns

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

Table 2 Resource Picker wildcard examples

Pattern	Use
*.sun.com	Matches any string ending with the characters .sun.com.
(products|docs).sun.com	Matches either products.sun.com or docs.sun.com.
198.93.9[23].???	Matches a numeric string starting with either 198.93.92 or 198.93.93 and ending with any 3 characters.
.	Matches any string with a period in it.
~sun-	Matches any string except those starting with sun-.
*.sun.com~docs.sun.com	Matches any host from domain sun.com except for a single host docs.sun.com.

Table 2 Resource Picker wildcard examples

Pattern	Use
`*.sun.com~(products|docs|software).sun.com`	Matches any host from domain `sun.com` except for hosts `products.sun.com`, `docs.sun.com`, and `software.sun.com`.
`*.com~*.sun.com`	Matches any host from domain com except for hosts from subdomain `sun.com`.

Administering Sun ONE Web Servers

This chapter describes how to administer Sun ONE Web Server 6.1 with the Sun ONE Web Server Administration Server. Using the Administration Server, you can manage servers, add and remove servers, and migrate servers from a previous release.

This chapter includes the following sections:

Starting the Administration Server

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

UNIX/Linux Platforms

To access the Administration Server in UNIX or Linux platforms:

1. Go to the *server_root*`/https-admserv/` directory (for example, `/usr/s1ws61/servers/https-admserv/`)

2. Type `./start`.

   This command starts the Administration Server using the port number you specified during installation.

Windows Platforms

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

- Release Notes
- Start Web Server Administration Server
- Uninstall Web Server
- Administer Web Server

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

To access the Administration Server on Windows platforms, perform the following steps:

1. Double-click the “Start Web Server Administration Server” icon, or type the following URL for starting the administration server in your browser:

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

   Sun ONE Web Server then displays a window prompting you for a user name and password.

2. Type the administration user name and password you specified during installation.

   Sun ONE Web Server displays the Administration Server page.

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

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

You can also access the Administration Server from a remote location as long as you have access to client software such as Netscape Navigator. Since the Administrator Server is accessed through a browser, you can access it from any machine that can reach the server over the network.

Running Multiple Servers

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

- Use virtual servers
- Install multiple instances of the server

Virtual Servers

Virtual servers allow you, with a single installed server, to offer companies or individuals domain names, IP addresses, and some server administration capabilities. For the users, it is almost as if they have their own web server, though you provide the hardware and basic web server maintenance.

The settings for virtual servers are stored in the `server.xml` file, found in the *server_root/server_id*`/config` directory. You do not need to edit this file to use virtual servers, but if you would like to learn more about this file, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference*.

For more information about virtual servers, see Chapter 13, "Using Virtual Servers."

Installing Multiple Instances of the Server

In past releases of Sun ONE Web Server, virtual servers did not have unique configuration information. The only way to have servers with separate configuration information was to create a new server instance. However, with Sun ONE Web Server 6.1, virtual servers have separate configuration information, so multiple server instances are no longer required. They are still supported, but virtual servers are the preferred way to have multiple servers.

If you choose to install multiple instances of the web server, you can use the Administration Server to:

- Install multiple copies of the server on Windows as separate instances, each with a different IP address.
- Configure a set of servers that all use the same IP address, but different port numbers.

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

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

To add another server instance, perform the following steps:

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

   Note that the server identifier cannot start with a digit and only Latin-1 characters should be used in instance names.
4. Click OK.

For more information, see the Add Server page in the online help.

Removing a Server

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

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

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

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

3. Select the server you wish to remove and click Yes.
4. Click OK.

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

server_root/https-*server-id*

For more information, se the Remove Server page in the online help.

Migrating a Server From a Previous Version

You can migrate a Sun ONE Web Serverfrom a 4.1 or 6.0 version to a 6.1 version. Your 4.1 or 6.0 server is preserved, and a new 6.1 server using the same settings is created.

You should stop running the 4.1 or 6.0 server before migrating settings. Make sure you have a compatible version of a web browser installed on your computer before migrating settings.

For a complete description of how to migrate a server from a previous version to Sun ONE Web Server 6.1, see the *Installation and Migration Guide.*

For more information, see the Migrate Server page in the online help.

Part 2

Using the Administration Server

Chapter 3

Managing Users and Groups

This chapter describes how to add, delete, and edit the users and groups who can access your Sun ONE Web Server.

This chapter includes the following sections:

Accessing Information About Users and Groups

The Administration Server provides access to your application data about user accounts, group lists, access privileges, organization 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 ONE Directory Server, which supports Lightweight Directory Access Protocol (LDAP). LDAP is an open directory access protocol that runs over TCP/IP and is scalable to a global size and millions of entries.

Since Sun ONE Web Server does not support local LDAP, you must have a directory server installed before you can add users and groups.

About Directory Services

A directory server such as Sun ONE Directory Server allows you to manage all your user information from a single source. You can also configure the directory server to allow your users to retrieve directory information from multiple, easily accessible network locations.

In Sun ONE Web Server 6.1, you can configure three different types of directory services to authenticate and authorize users and groups. If no other directory service is configured, the new directory service created will be set to the value `default`, irrespective of its type.

When you create an a directory service, the *`server-root`*`/userdb/dbswitch.conf` file is updated with the directory service details.

Types of Directory Services

The different types of directory services supported by Sun ONE Web Server 6.1 are:

- **LDAP.** Stores user and group information 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://draco.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://draco.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcom

  ldap:binddn cn=Directory Manager

  ldap:encoded bindpw YWRtaW5hZG1pbg==
  ```

- **Key File.** A keyfile 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 users and groups stored in a keyfile are used for authorization and authentication by the `file` realm alone; these bear no relationship to system users and groups. For more information about the `file` realm, see "File realm."

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

  ```
  directory keyfile file

  keyfile:syntax keyfile

  keyfile:keyfile D:\draco\keyfile\keyfiledb
  ```

- **Digest File.** Stores user and group information based on encrypted username and password.

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

  ```
  directory digest file

  digest:syntax digest

  digest:digestfile D:\draco\digest\digestdb
  ```

NOTE If you want to set up distributed administration, the default directory service must be an LDAP-based directory service.

Configuring a Directory Service

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

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Configure Directory Service link.
3. From the Create New Service of Type drop-down list, choose the type of directory service you want to create.

4. Click New.

   You can now configure the directory service information in the page corresponding to the type of directory service you have selected.

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

5. Click Save Changes to save your changes.

Once you create and configure directory services, you can assign directory services per virtual server. The rights and permissions associated with the directory service is later used by the server to evaluate and enforce access control rules. For more information, see "Choosing a Directory Service for a Virtual Server."

Understanding Distinguished Names (DNs)

Use the Users and Groups tab of the Administration Server to create or modify users, groups, and organizational units. A user is an individual in your 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 company that uses the `organizationalUnit` object class. Users, groups, and organizational units are described further 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 you make changes to a user or group directory entry. For example, you need to specify DN information each time you create or modify directory entries, set up access controls, and set up user accounts for applications such as mail or publishing. The users and groups interface of the Sun ONE Web Server Administration Console helps you 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 before each equal sign in this example have the following meanings:

- `uid`: user ID

- e: email address
- cn: the user's common name
- o: organization
- c: country

DNs may include a variety of name-value pairs. They 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 if you want to add a new subtree to an existing directory, you can use the Directory Server's Administration Server LDIF import function. This function accepts a file containing LDIF and attempts to build a directory or a new subtree from the LDIF entries. You can also export your current directory to LDIF using the Directory Server's LDIF export function. This function creates an LDIF-formatted file that represents your directory. Add or edit entries using the ldapmodify command 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 Directory Server.

Creating Users

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

When you create a user, you must protect server security by ensuring that the user does not have unauthorized access to resources. Sun ONE Web server 6.1 provides you with a range of choices to enhance security:

- For information on how to use J2EE / Servlet-based realm authentication to authenticate and authorize users, see "Realm-based Security" on page 89.
- For information on how to use Access Control List (ACL)-based authorization and authentication techniques, see "How Access Control Works" on page 189.
- For information on using the Native Realm functionality that bridges the Java-based security model and the ACL-based security model, see "Configuring the Native Realm" on page 95.

This section includes the following topics:

Creating a New User in an LDAP-based Authentication Database

When you add user entries to an LDAP-based directory service, the services of an underlying LDAP-based directory server are used to authenticate and authorize users. This section provides certain guidelines you need to consider while using an LDAP-based authentication database and describes how you can add users through the Administration Server.

Guidelines for Creating LDAP-based User Entries

Consider the following guidelines when using the administrator forms to create new user entries in an LDAP-based directory service:

- If you enter a given name (or first name) and a surname, then the form automatically fills in the user's full name and user ID for you. The user ID is generated as the first initial of the user's first name followed by the user's last name. For example, if the user's name is Billie Holiday, then the user ID is automatically set to bholiday. You can replace this user ID with an ID of your own choosing if you wish.
- The user ID must be unique. The Administration Server ensures that the user ID is unique by searching the entire directory from the search base (`base DN`) down to see if the user ID is in use. Be aware, however, that if you use the Directory Server ldapmodify command line utility (if available) to create a user, that it does not ensure unique user IDs. If duplicate user IDs exist in your directory, the affected users will not be able to authenticate to the directory.

- Note that the base DN specifies the distinguished name where directory lookups will occur by default, and where all Sun ONE Web Administration Server's entries are placed in your directory tree. A "DN" is the string representation for the name of an entry in a directory server.
- Note that at a minimum, you must specify the following user information when creating a new user entry:
  - surname or last name
  - full name
  - user ID
- If any organizational units have been defined for your directory, you can specify where you want the new user to be placed using the Add New User To list. The default location is your directory's base DN (or root point).

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

How to Create a New User Entry

To create a user entry, read the guidelines outlined in "Guidelines for Creating LDAP-based User Entries" on page 58, then perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the New User link.
3. Select the LDAP Directory Service from the Select Directory service drop-down list, and click Select.
4. Add the required information to the page that comes up.

   For more information see Directory Server User Entries.
5. Click OK.

For more information, see the New User page in the online help.

Directory Server User Entries

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

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

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

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

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

  However, note that you can change this format to a uid-based distinguished name.
- The values on the user form fields are stored as the following LDAP attributes (note that any stored information other than 'user' and 'group' requires a full Directory Server license):

Table 3-1 LDAP Attributes

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

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

Table 3-2 User Entry LDAP Attributes

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

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

Creating a New User in a File-based Authentication Database

Sun ONE Web Server 6.1 introduces support for native authentication databases that store user information in flat files in text format. The file-based authentication database is compatible with the following types of files:

- keyfile-style files
- digest-style files
- .htaccess-style files

Creating a New User Entry

To create a user entry in a file-based authentication database, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the New User link.
3. Select the file-based directory service ID from the Select Directory service drop-down list and click Select.
4. Enter the following information:
    - **User ID.** (Required) Specifies a unique user name for the user.
    - **Password.** Specifies the password for the user.
    - **Password (again).** Confirms the password entered in the Password field.
    - **Groups.** Specifies a comma-separated list of groups of which the user is a member.
5. Click Create User.

Creating a New User in a Digest-based Authentication Database

To create a user entry in a digest-based authentication database, which stores user and group information in an encrypted form, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the New User link.
3. Select the digest-based directory service ID from the Select Directory Service drop-down list and click Select.
4. Enter the following information:
   - **User ID.** (Required) Specifies a unique user name for the user.
   - **Realm.** Specifies the realm that will authenticate this user.
   - **Password.** Specifies the password for the user.
   - **Password (again).** Confirms the password entered in the Password field.
   - **Groups.** Specifies a comma-separated list of groups of which the user is a member.
5. Click OK.

Managing Users

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

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

This section includes the following topics:

- Managing User Licenses
- Renaming Users
- Removing Users

Finding User Information

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

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Users link.
3. In the Find User field, enter some descriptive value for the entry that you want to edit. You can enter any of the following in the search field:
   - A name. Enter a full name or a partial name. All entries that equally match the search string will be returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
   - A user ID.
   - A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number will be returned.
   - An email address. Any search string containing an at (@) symbol is assumed to be an email address. If an exact match cannot be found, then a search is performed to find all email addresses that begin with the search string.
   - An asterisk (*) to see all of the entries currently in your directory. You can achieve the same effect by simply leaving the field blank.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered a search filter.

   As an alternative, use the drop-down menus in the Find all users whose field to narrow the results of your search.
4. In the Look within field, select the organizational unit under which you want to search for entries.

   The default is the directory's root point (or top most entry).
5. In the Format field, choose either On-Screen or Printer.

6. Click Find.

   All the users in the selected organizational unit are displayed.

7. In the resulting table, click the name of the entry that you want to edit.

   The user edit form is displayed.

8. Change the displayed fields as desired and click Save Changes.

   The changes are made immediately.

Building Custom Search Queries

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

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

- The left-most drop-down list allows you to specify the attribute on which the search will be based.

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

Table 3-3 Search Attribute Options

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

- In the center drop-down list, select the type of search you want to perform.

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

Table 3-4 Search Type Options

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

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

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

Editing User Information

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

1. Access the Administration Server and choose the Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 63.
3. Edit the field corresponding to the attribute that you wish to change.

For more information, see the Edit Users page in the online help.

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

In addition, note that you can change the user's first, last, and full name field from this form, but to fully rename the entry (including the entry's distinguished name), you need to use the Rename User form. For more information on how to rename an entry, see "Renaming Users," on page 67.

Managing a User's Password

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

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

1. Access the Administration Server and choose Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 63.
3. Make the desired changes and click OK.

For more information, see the Manage Users page in the online help.

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

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

Managing User Licenses

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

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

1. Access the Administration Server and choose the Users & Groups tab.
2. Display the user entry as described in "Finding User Information," on page 63.
3. Click the Licenses link at the top of the User Edit form.
4. Make the desired changes and click OK.

For more information, see the Manage Users page in the online help.

Renaming Users

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

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

To rename a user entry, perform the following steps:

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

2. Display the user entry as described in “Finding User Information,” on page 63.

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

3. Click the Rename User button.

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

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

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

For more information, see the Manage Users page in the online help.

Removing Users

To delete a user entry, perform the following steps:

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

2. Display the user entry as described in “Finding User Information,” on page 63.

3. Click Delete User.

For more information, see the Manage Users page in the online help.

Creating Groups

A group is an object that describes a set of objects in an LDAP database. An Sun ONE Web 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.

There are two ways to define membership of a group: statically and dynamically. Static groups enumerate their member objects explicitly. A static group is a CN and contains `uniqueMembers` and / or `memberURLs` and / or `memberCertDescriptions`. For static groups, the members do not share a common attribute except for the `CN=<Groupname>` attribute.

Dynamic groups allow you to use a LDAP URL to define a set of rules that match only for group members. For Dynamic Groups, the members do share a common attribute or set of attributes that are defined in the `memberURL` filter. For example, if you need a group that contains all employees in Sales, and they are already in the LDAP database under

"`ou=Sales,o=Airius.com`," you'd define a dynamic group with the following `memberurl`:

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

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

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

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

This section includes the following topics:

Static Groups

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

Guidelines for Creating Static Groups

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

- Static groups can contain other static or dynamic groups.
- You can optionally also add a description for the new group.
- If any organizational units have been defined for your directory, you can specify where you want the new group to be placed using the Add New Group To list. The default location is your directory's root point, or top-most entry.

- When you are finished entering the desired information, click Create Group to add the group and immediately return to the New Group form. Alternatively, click Create and Edit Group to add the group and then proceed to the Edit Group form for the group you have just added. For information on editing groups, see “Editing Group Attributes,” on page 75.

To Create a Static Group

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

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

For more information, see the New Group page in the online help.

Dynamic Groups

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

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

This section includes the following topics:

How Sun ONE Web Server Implements Dynamic Groups

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

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

This example describes a set that consists of all objects below "`o=mcom.com`" whose department is "marketing." The LDAP URL can contain a search base DN, a scope and filter, however, not a hostname and port. This means that you can only refer to objects on the same LDAP server. All scopes are supported.

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

NOTE Sun ONE Web Server uses the `cn (commonName)` attribute as group name for ACLs.

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

Groups Can Be Static and Dynamic

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

Dynamic Group Impact on Server Performance

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

Guidelines for Creating Dynamic Groups

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

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

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

  The required parameters are described in the following table:

Table 3-5 Dynamic Groups: Required Parameters

Parameter Name	Description
`<base_dn>`	The Distinguished Name (DN) of the search base, or point from which all searches are performed in the LDAP directory. This parameter is often set to the suffix or root of the directory, such as "`o=mcom.com`".
`<attributes>`	A list of the attributes to be returned by the search. To specify more than one, use commas to delimit the attributes (for example, "`cn,mail,telephoneNumber`"); if no attributes are specified, all attributes are returned. Note that this parameter is ignored for dynamic group membership checks.

Table 3-5 Dynamic Groups: Required Parameters

Parameter Name	Description
<scope>	The scope of the search, which can be one of these values: • **base** retrieves information only about the distinguished name (<base_dn>) specified in the URL. • **one** retrieves information about entries one level below the distinguished name (<base_dn>) specified in the URL. The base entry is not included in this scope. • **sub** retrieves information about entries at all levels below the distinguished name (<base_dn>) specified in the URL. The base entry is included in this scope. This parameter is required.
<(filter)>	Search filter to apply to entries within the specified scope of the search. If you are using the Administration Server forms, you must specify this attribute. Note that the parentheses are required. This parameter is required.

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

- You can optionally also add a description for the new group.
- If any organizational units have been defined for your directory, you can specify where you want the new group to be placed using the Add New Group To list. The default location is your directory's root point, or top-most entry.
- When you are finished entering the desired information, click Create Group to add the group and immediately return to the New Group form. Alternatively, click Create and Edit Group to add the group and then proceed to the Edit Group form for the group you have just added. For information on editing groups, see "Editing Group Attributes" on page 75.

To Create a Dynamic Group

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

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the New Group link.

3. Select Dynamic Group from the Type of Group drop-down list.
4. Enter the required information and click OK.

For more information, see the New Group page in the online help.

Managing Groups

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

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

Finding Group Entries

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

To find a group entry, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link.
3. Enter the name of the group that you want to find in the Find Group field.

   You can enter any of the following values in the search field:

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

As an alternative, use the drop-down menus in “Find all groups whose” to narrow the results of your search.

4. In the Look within field, select the organizational unit under which you want to search for entries.

   The default is the directory’s root point, or top-most entry.

5. In the Format field, choose either On-Screen or Printer.
6. Click Find.

   All the groups matching your search criteria are displayed.

7. In the resulting table, click the name of the entry that you want to edit.

The “Find all groups whose” Field

The “Find all groups whose” field allows you to build a custom search filter. Use this field to narrow down the search results that are otherwise returned by Find groups.

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

For more information regarding how to build a custom search filter, see “Building Custom Search Queries,” on page 64.

Editing Group Attributes

To edit a group entry, perform the following steps:

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

For more information regarding how to find specific entries, refer to the concepts outlined in “Finding Group Entries,” on page 74.

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

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

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

Adding Group Members

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

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link.
3. Locate the group you want to manage as described in “Finding Group Entries,” on page 74, and click the Edit button under Group Members.

   Sun ONE Web Server displays a new form that enables you to search for entries. If you want to add user entries to the list, make sure Users is shown in the Find drop-down list. If you want to add group entries to the group, make sure Group is shown.
4. In the right-most text field, enter a search string. Enter any of the following options:
   - A name. Enter a full name or a partial name. All entries whose name matches the search string is returned. If no such entries are found, all entries that contain the search string are found. If no such entries are found, any entries that sounds like the search string are found.
   - A user ID if you are searching for user entries.

- A telephone number. If you enter only a partial number, any entries that have telephone numbers ending in the search number are returned.
- An email address. any search string containing an at (@) symbol is assumed to be an email address. If an exact match cannot be found, then a search is performed to find all email addresses that begin with the search string.
- Enter either an asterisk (*) or simply leave this text field blank to see all of the entries or groups currently residing in your directory.
- Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.

5. Click Find and Add to find all the matching entries and add them to the group.

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

6. When the list of group members is complete, click Save Changes.

   The currently displayed entries are now members of the group.

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

Adding Groups to the Group Members List

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

To add a group to the members list of another group, add the group as if it were a user entry. For more information, see "Adding Group Members," on page 76.

Removing Entries from the Group Members List

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

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

2. Click the Manage Groups link, locate the group you want to manage as described in “Finding Group Entries,” on page 74, and click the Edit button under Group Members.
3. For each member that you want to remove from the list, click the corresponding box under the Remove from list? column.

   Alternatively, you can construct a filter to find the entries you want to remove and click the Find and Remove button. For more information on creating a search filter, see “Adding Group Members,” on page 76.
4. Click Save Changes. The entry(s) are deleted from the group members list.

Managing Owners

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

Table 3-6 Additional Information

Task You Want to Complete	Read Section
Add owners to the group	“Adding Group Members,” on page 76.
Add groups to the owners list	“Adding Groups to the Group Members List,” on page 77.
Remove entries from the owners list	“Removing Entries from the Group Members List,” on page 77.

Managing See Alsos

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

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

Table 3-7 Additional Information

Task You Want to Complete	Read Section
Add users to see alsos	“Adding Group Members,” on page 76.

Table 3-7 Additional Information

Task You Want to Complete	Read Section
Add groups to see alsos	"Adding Groups to the Group Members List," on page 77.
Remove entries from see alsos	"Removing Entries from the Group Members List," on page 77.

Removing Groups

To delete a group, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link, locate the group you want to manage as described in "Finding Group Entries," on page 74, and click Delete Group.

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

Renaming Groups

To rename a group, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Groups link and locate the group you want to manage as described in "Finding Group Entries," on page 74.
3. Click the Rename Group button and type the new group name in the resulting dialog box.

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

- organizational units for Marketing and Product Management
- a group named Online Sales under the Marketing organizational unit

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

Creating Organizational Units

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

To create an organizational unit, perform the following steps:

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

For more information, see the New Organizational Unit page in the online help.

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

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

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

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

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

Managing Organizational Units

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

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

- Deleting Organizational Units

Finding Organizational Units

To find organizational units, perform the following steps:

1. Access the Administration Server and choose the Users & Groups tab.
2. Click the Manage Organizational Units link.
3. Type the name of the unit you want to find in the Find organizational unit field. You can enter any of the following in the search field:
   - A name. Enter a full name or a partial name. All entries that equally match the search string will be returned. If no such entries are found, all entries that contain the search string will be found. If no such entries are found, any entries that sounds like the search string are found.
   - An asterisk (*) to see all of the groups currently residing in your directory. You can achieve this same result by simply leaving the field blank.
   - Any LDAP search filter. Any string that contains an equal sign (=) is considered to be a search filter.

   As an alternative, use the drop-down menus in the Find all units whose field to narrow the results of your search.
4. In the Look within field, select the organizational unit under which you want to search for entries.

   The default is the root point of the directory.
5. In the Format field, choose either On-Screen or Printer.
6. Click Find.

   All the organizational units matching your search criteria are displayed.
7. In the resulting table, click the name of the organizational unit that you want to find.

The “Find all units whose” Field

The Find all units whose field allows you to build a custom search filter. Use this field to narrow down the search results that are otherwise returned by Find organizational unit.

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

For more information regarding how to build a custom search filter, see “Building Custom Search Queries,” on page 64.

Editing Organizational Unit Attributes

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

1. Locate the organizational unit you want to edit as described in “Finding Organizational Units,” on page 81.

   The organizational unit edit form is displayed.

2. Change the displayed fields as desired and click Save Changes.

   The changes are made immediately.

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

Renaming Organizational Units

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

1. Make sure no other entries exist in the directory under the organizational unit that you want to rename.
2. Locate the organizational unit you want to edit as described in “Finding Organizational Units,” on page 81.
3. Click the Rename button.
4. Enter the new organizational unit name in the resulting dialog box.

NOTE When you rename an organizational unit entry, you can only change the organizational unit’s name; you cannot use the rename feature to move the entry from one organizational unit to another. For more information, see “Renaming Organizational Units,” on page 82.

Deleting Organizational Units

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

1. Make sure no other entries exist in the directory under the organizational unit that you want to rename.
2. Locate the organizational unit you want to delete as described in “Finding Organizational Units,” on page 81.
3. Click the Delete button.
4. Click OK in the resulting confirmation box.

   The organizational unit is immediately deleted.

Chapter 4

J2EE-based Security for Web Container and Web Applications

This chapter describes the basic features of J2EE-based security for the Sun ONE Web Server 6.1 web container and web applications. It begins with a discussion on the two main authentication and authorization models supported by the Web server : the Access Control List (ACL)-based security model and the J2EE / Servlet-based security model. It also discusses new functionality in Sun ONE Web Server 6.1 that allows you to deploy Java web applications which can leverage the benefits of both security systems.

While the rest of this chapter deals with the J2EE / Servlet configuration issues, related security issues are described in the following chapters:

- Certificates and public key cryptography, in Chapter 6, “Using Certificates and Keys”.
- ACL-based security, in Chapter 9, “Controlling Access to Your Server”.

This chapter contains the following sections:

About Sun ONE Web Server Security

You can protect resources that reside on your Web server through several security services and mechanisms, including authentication, authorization, and access control.

Authentication is the process of confirming an identity. Authorization means granting access to a restricted resource to an identity, and access control mechanisms enforce these restrictions. Authentication and authorization can be enforced by a number of security models and services.

Sun ONE Web Server 6.1 supports two security models: the ACL-based security model provided by the HTTP engine and the J2EE Servlet version 2.3 specification -based provided by the web container.

Both models co-exist in the life time of a Sun ONE Web Server 6.1 process. Each model supports both client authentication and authorization security services.

The Sun ONE Web Server 6.1 web container provides client authentication through the Java Authentication and Authorization Service (JAAS)-based realm mechanism, and authorization through the J2EE role-based mechanism. One of the realms provided by Sun ONE Web Server 6.1 is the Native Realm. It provides the bridge between the two security models.

Sun ONE Web Server 6.1 supports both declarative security and programmatic security.

Sun ONE Web Server 6.1 leverages the features of the J2EE platform to define declarative contracts between those who develop and assemble application components and those who configure applications in operational environments. In the context of application security, application providers are required to declare the security requirements of their applications in such a way that these requirements can be satisfied during application configuration. The declarative security mechanisms used in an application are expressed in a declarative syntax in a document called a *deployment descriptor*. An application deployer then employs container-specific tools to map the application requirements that are in a deployment descriptor to security mechanisms that are implemented by J2EE containers. The deployment descriptor files for web applications in Sun ONE Web Server 6.1 are the `web.xml` and `sun-web.xml` files.

Programmatic security refers to security decisions that are made by security-aware applications. Programmatic security is useful when declarative security alone is not sufficient to express the security model of an application. For example, an application might make authorization decisions based on the time of day, the parameters of a call, or the internal state of a web component. Another application might restrict access based on user information stored in a database.

The rest of this chapter runs you through the following key concepts in authentication and authorization supported by Sun ONE Web Server 6.1:

- ACL-based access control, described in the section Overview of ACL-based Access Control.
- J2EE-based access control, described in the section Overview of J2EE / Servlet-based Access Control.
- Native Realm support, described in the section Native Realm.
- Programmatic security, described in the section Using Programmatic Security.

Overview of ACL-based Access Control

ACL-based access control is described at length in Chapter 9, “Controlling Access to Your Server”. The following section provides a brief overview of the key concepts.

Sun ONE Web Server 6.1 supports authentication and authorization through the use of locally stored access control lists (ACLs), which describe what access rights a user has for a resource. For example, an entry in an ACL can grant a user named John `read` permission to a particular folder, `misc`.

```
acl "path=/export/user/990628.1/docs/misc/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all) (user=”anyone”);
  allow (read) (user = "John");
```

The core ACLs in Sun ONE Web Server 6.1 support three types of authentication: basic, SSL, and digest.

Basic authentication relies on lists of user names and passwords passed as cleartext. The SSL method requires the browser to have a user certificate, which contains the user’s public key and other user information such as name, email, and so on. Digest authentication uses encryption techniques to encrypt the user’s credentials.

The main features of the ACL-based access control model are described below:

- ACL-based authentication and authorization use the following configuration files:
  - *server-install*/httpacl/*.acl files
  - *server-install*/userdb/dbswitch.conf
  - *server-install*/*server-instance*/config/server.xml
- Authentication databases are provided by auth-db modules which are configured in the dbswitch.conf file.
- Authentication and authorization is performed by access control rules set in the *server-install*/httpacl/*.acl files, if ACLs are configured. The authorization rules that apply are those that are defined in the ACL file corresponding to the virtual server processing the request (as configured in the appropriate VS entry in server.xml) See the ACLFILE element and aclids property of the VS element in the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference*. Typically these files are located in the /httpacl/ directory, but don't necessarily have to be if you change the server.xml configuration.

In addition, the Sun ONE Web Server 6.1 SSL engine supports external crypto hardware to offload SSL processing and to provide optional tamper-resistant key storage.

For more information about access control and the use of external crypto hardware, see Chapter 9, "Controlling Access to Your Server".

Overview of J2EE/Servlet-based Access Control

J2EE / Servlet-based access control is described at length in the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*. The following section provides a brief overview of the key concepts.

Sun ONE Web Server 6.1, apart from providing ACL-based authentication, also leverages the security model defined in the J2EE 1.3 Specification to provide several features that help you develop and deploy secure Java Web applications.

A typical J2EE-based Web application consists of the following parts, access to any or all of which can be restricted:

- Servlets
- JavaServer Pages (JSP) components

- HTML documents
- Miscellaneous resources, such as image files and compressed archives

The J2EE / Servlet-based access control infrastructure relies on the use of security realms. When a user tries to access an access-protected section of an application through a Web browser, the Web container prompts for the user's credential information, and then passes it for verification to the realm that is currently active in the security service for this particular application.

The main features of the J2EE / Servlet-based access control model are described below:

- J2EE / Servlet-based authentication uses the following configuration files:
  - The web application deployment descriptor files `web.xml` and `sun-web.xml`
  - *server-install*/*server-instance*/`config/server.xml`
- Authentication is performed by Java security realms which are configured through `AUTHREALM` entries in the `server.xml` file.
- Authorization is performed by access control rules in the deployment descriptor file, `web.xml`, in case any such rules have been set.

The following section briefly explains the concept of security realms. For a fuller discussion on the J2EE security model and realm-based authentication, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*.

Realm-based Security

The J2EE-based security model provides for security realms that identify and authenticate users. The user information is obtained from an underlying security realm. Realm-based security consists of two aspects:

- Realm-based User Authentication. This verifies users through an underlying realm.
- Role-based Authorization. This assigns users to roles, which in turn are granted or restricted access to resources.

Realm-based User Authentication

The authentication process verifies users through an underlying realm, also known as a security domain. A realm consists of a set of users, optional group mappings, and authentication logic that can validate authentication requests. Once an authentication request is validated by a configured realm and the security context established, this identity is applied to all subsequent authorization decsisions, unless overruled by a `run-as` condition.

A server instance may have any number of configured realms. The configuration information is present in the `AUTHREALM` element in the `server.xml` file.

In Sun ONE Web Server, the authentication service is built using JAAS, which provides pluggable security domains. The Java authentication realms in Sun ONE Web Server 6.1 are compatible with Sun ONE Application Server 7.0 realms.

Sun ONE Web Server 6.1 provides the following realms:

- LDAP realm
- File realm
- Solaris realm
- Certificate realm
- Custom Realm
- Native Realm

LDAP realm

The `ldap` realm allows you to use an LDAP database for user security information. An LDAP directory service is a collection of attributes with unique identifiers. The `ldap` realm is ideal for deployment to production systems.

In order to authenticate users against the `ldap` realm, you must create the desired user(s) in your LDAP directory. You can do this from the Administration Server's Users & Groups tab or from your LDAP directory product's user management console. For more information, see "Creating a New User in an LDAP-based Authentication Database" on page 58.

File realm

The `file` realm is the default realm when you first install the Sun ONE Web Server. This realm, easy and simple to set up, represents a significant convenience to developers.

The `file` realm authenticates users against user data stored in a text file. The following authentication databases are supported by the file realm:

- keyfile-style databases
- htaccess-style databases
- digest-style databases

For more information about the various file-based authentication databases, see <add>.

The user information file used by the `file` realm is initially empty, so you must add users before you can use the `file` realm. For more information on how you can do this, see "Creating a New User in a File-based Authentication Database" on page 61.

Solaris realm

The `solaris` realm allows authentication using Solaris username + password data. This realm is only supported on Solaris 9. Because this realm uses the user database in the Solaris 9 Operating Environment, it eliminates the extra step of setting up a separate database.

Certificate realm

The certificate realm supports SSL authentication. The certificate realm sets up the user identity in the Sun ONE Web Server's security context and populates it with user data from the client certificate. The J2EE containers then handle authorization processing based on each user's DN from his or her certificate. This realm authenticates users with SSL or TLS client authentication through X.509 certificates.

For details on how to set up the server and client certificates, see Chapter 6, "Using Certificates and Keys".

Custom Realm

You can build realms for other databases, such as Oracle, to suit your specific needs by using pluggable JAAS login modules and a realm implementation. Note that client-side JAAS login modules are not suitable for use with Sun ONE Web Server.

Refer to the sample realm in Sun ONE Web Server 6.1 as a template.

Native Realm

The Native realm is a special realm that provides a bridge between the core ACL-based authentication model and the J2EE / Servlet authentication model. By using the Native realm for Java web applications it becomes possible to have the ACL subsystem perform the authentication (instead of having the Java web container do so) and yet have this identity available for Java web applications.

When an authentication operation is invoked, the Native realm delegates this authentication to the core authentication subsystem. From the user's perspective this is essentially equivalent to, for example, the LDAP realm delegating authentication to the configured LDAP server. When group membership queries are processed by the Native realm, they are also delegated to the core authentication subsystem. From the Java web modules and the developers perspective, the Native realm is no different from any of the other Java realms which are available for use with web modules.

Since Native realm delegates the authentication to the core, some additional configuration is required. For more information, see Configuring the Native Realm.

The Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications* provides a detailed discussion on J2EE security realms and the configuration parameters you can use to configure security realms.

Role-based Authorization

The Java Servlet 2.3 Specification define how to establish access control rules to restrict access to the various J2EE application resources.

Mapping Roles to Restricted Areas

J2EE access control is based on roles. To restrict access to specific HTML pages, servlets, JSPs, and so forth, you must define the following:

- The restricted areas, as listed in the Web module descriptors (`web.xml`)
- The roles which are granted access to each restricted area (in `web.xml`)
- User and group mappings to roles, that determine which specific users are authorized to access which restricted areas (in `sun-web.xml`).

Users can assume multiple roles and, upon verification that they have been assigned at least one of the roles, they are allowed access to the corresponding areas.

Use the samples located in the `webapps/security` directory with various access restrictions in Sun ONE Web Server 6.1 as templates. For additional discussion on Servlet role-based security, refer to the Servlet 2.3 specification.

Defining Access Control by Roles

J2EE application roles are abstract ones and apply to specific applications. To run your application in a real-world environment with restricted access to authorized users only, you must map the user names to the roles in the `sun-web.xml` descriptor. Employ either or both of these ways:

Principal mapping - Map a user name or multiple names directly to a role in `sun-web.xml`. This method is convenient for testing but does not scale beyond a limited number of users in each role.

Group mapping - Map a user name or multiple ones indirectly through one or multiple groups in `sun-web.xml`. (For example, group names can be engineers, managers, or staff.) Any authenticated user who belongs to the groups listed is then assigned the application role. Please note that the active realm implementation (or the database that is references) is responsible for determining which users belong to a given group.

When a principal (user) requests a particular Web resource, for example, a servlet or a JSP, the Web container checks the security constraints or permissions associated with the resource in the deployment descriptor files to determine whether the principal is authorized to access it.

Role mapping entries map a role to a user or a group in the module descriptor. Example:

```
<sun-web-app>
    <security-role-mapping>
    <role-name>manager</role-name>
    <principal-name>jsmith</principal-name>
    <group-name>divmanagers</group-name>
</sun-web-app>
```

For more information about deployment descriptor files, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*.

How to Configure a Realm

You can configure realms in one of these ways:

Using the Administration Interface

To configure a realm using the Administration interface:

1. From the Administration Server interface, access the server instance you want to manage, then click on the Java tab.
2. Click the Security Realms link.

   By default, the following realms are provided:

   - file
   - native
   - ldap
3. To add a realm, click the New button. To delete a realm, check the checkbox next to the name of the realm, and click OK. To edit a realm, click onthe name of the realm.
4. If you are adding or editing a realm, enter the realm's name, classname, properties, and users (`file` realm only), then click the OK button.
5. Click OK.

Editing the server.xml File

Behind the scenes, the default realm is set in the `SECURITY` element in the `server.xml` file. The `SECURITY` configuration looks like this:

```
<SECURITY defaultrealm="file" anonymousrole="ANYONE"
      audit="false">
   <AUTHREALM name="file"
         classname="com.iplanet.ias.security.auth.realm.file.FileRe
alm">
      <property name="file" value="instance_dir/config/keyfile"/>
```

```
      <property name="jaas-context" value="fileRealm"/>
   </AUTHREALM>
   ...
</SECURITY>
```

The `defaultrealm` attribute points to the realm the server is using by default. The default realm will be used by all web applications which do not provide a valid realm in their `web.xml`. It must point to one of the configured `AUTHREALM` names. The default is the `file` realm.

The `audit` flag determines whether auditing information is logged. If set to `true`, the server logs audit messages for all authentication and authorization events.

If you change the realm configuration, you must restart the server for the change to take effect.

For more information about the `server.xml` file, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference*.

Configuring the Native Realm

As with all realms, you can configure the Native realm using the `AUTHREALM` element within the `SECURITY` element in `server.xml`. Example:

```
<AUTHREALM name="native"
classname="com.sun.enterprise.security.auth.realm.webcore.NativeRea
lm">
   <PROPERTY name="auth-db" value="mykeyfile">
   <PROPERTY name="jaas-context" value="nativeRealm"/>
</AUTHREALM>
```

The `auth-db` property points to the core authentication database to which this Native realm instance will delegate all authentication requests; in this example, an authentication database named “mykeyfile”. This property is optional. If not specified, the core authentication engine will use the `default` auth-db to process all requests from this Native realm. As in the case of most realms, the `jaas-context` property is a pointer to the JAAS login context to be used (defined in `login.conf`).

No other configuration is needed by the Native realm. However, since requests are being delegated to a core authentication database, that particular authentication database must also be property configured. The rest of this section provides an example of configuring a core authentication database.

To configure a core (native) authentication database, in `server.xml`, the `VS` element must contain a `USERDB` element which maps the `auth-db` name to a database name. For example:

```
<VS id="https-plaza.com"  ....
....
   <USERDB id="mykeyfile" database="myalt"/>
....
</VS>
```

Note that if the `auth-db` property is not given (in which case "`default`" is used) you could have a `USERDB` entry mapping id="default" to some database name. If no mapping is present, the mapping is to `default`.

Next, the file *`install-root`*`/userdb/dbswitch.conf` must contain the configuration for the `myalt` database. The following example defines `myalt` to be a file-based authentication database.

```
directory myalt file
myalt:syntax keyfile
myalt:keyfile /local/ws61/https-plaza.com/config/keyfile
```

The above configuration is in no way specific to the Native realm. Any valid authentication directory configuration can be used as the destination authentication database by the Native realm. This means the Native realm can be configured to delegate to native LDAP authentication databases or even to custom native authentication databases.

NOTE In Sun ONE web Server 6.1, web applications have two distinct mechanisms for using LDAP as the authentication engine:

- Using the Java LDAP realm
- Using the Java Native realm configured to delegate to the native LDAP authentication database.

Specifying the Default Realm

The default realm is used to process authentication events for all web applications which do not specify a valid alternate realm in their `web.xml` deployment descriptor file. To specify the active authentication realm for the server instance, perform the following steps:

1. Access the Server Manager and choose the Java tab.
2. Click the Java Security link.
3. Set the following information:
   - **Default Realm** . Specifies the active authentication realm (an `AUTHREALM` name attribute) for this server instance.
   - **Anonymous Role** (optional). Used as the name for default or anonymous role.
   - **Audit Enabled** (optional). If true, additional access logging is performed to provide audit information. Audit information consists of:
     - Authentication success and failure events
     - Servlet access grants and denials
   - **Log Level** (optional). Controls the type of messages logged to the errors log.
4. Click OK.

Using Programmatic Security

In addition to the container-managed authentication provided by the realms, Sun ONE Web Server 6.1 also supports managed authentication accessed through the programmatic login interface. This interface supports custom authentication models that do not fit into the realm infrastructure. Programmatic login can also be used by J2EE applications to directly establish authentication contexts for themselves. However, such a practice makes the application less portable and less maintainable and is not recommended.

The `ProgrammaticLoginPermission` permission is required to invoke the programmatic login mechanism for an application. This permission is not granted by default to deployed applications because this is not a standard J2EE mechanism.

Sun ONE web Server 6.1 supports the Security Manager. The Security manager is disabled by default when you first install the server. If you have enabled the Java Security Manager in your server instance, you need to grant this permission to any web applications that will use programmatic login.

To grant the required permission to the application, you need to edit the `server.policy` file.

You can enable policy support by specifying the standard Java policy entries in the `server.xml` file:

```
<JVMOPTIONS>-Djava.security.manager</JVMOPTIONS>
```

```
<JVMOPTIONS>-Djava.security.policy=install-root/https-servername/config/server.policy</JVMOPTIONS>
```

For more information details about the `server.policy` file, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications.*

Deciding When to Use the J2EE/Servlet Authentication Model

This section is designed to help you understand in what circumstances you should decide to use the J2EE / Servlet-based authentication model.

Use the J2EE / Servlet authentication model:

- In general, for most new J2EE / Servlet-based web applications.
- For existing `.war` files that you do not wish to modify.
- For creating web applications where full J2EE / Servlet compatibility is important either now or in the future.
- If you wish to use form-based authentication since form-based authentication is not supported by ACLs.

Remember that even if you use the ACL-based infrastructure you still have the option of using the Native Realm Java realm in order to propagate the user identity so that it is available for the servlet.

Setting Administration Preferences

You can configure your Administration Server using the pages on the Preferences and Global Settings tabs. Note that you must enable cookies in your browser to run the CGI programs necessary for configuring your server.

This chapter includes the following sections:

Shutting Down the Administration Server

Once the server is installed, it runs constantly, listening for and accepting HTTP requests. You might want to stop and restart your server if, for instance, you have just installed a Java Development Kit (JDK) or Directory Server, or if you have changed listen socket settings.

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

- Access the Administration Server, choose the Preferences tab, select the Shut Down link, and click “Shut down the administration server!” button.

For more information, see the Shut Down page in the online help.

- Use the Services window in the Control Panel (Windows).
- Use `stop`, which shuts down the server completely, interrupting service until it is restarted.

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."

Editing Listen Socket Settings

Before the server can process a request, it must accept the request via a listen socket, then direct the request to the correct virtual server. When you install Sun ONE Web 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 HTTP server port number during installation (the default is 8888). You cannot delete the default listen socket.

You can edit your server's listen socket settings using the Administration Server's Listen Sockets Table. To access the table, perform the following steps:

1. Access the Administration Server and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Make the desired changes and click OK.

For more information, see Chapter 13, "Using Virtual Servers" and the online help for the Edit Listen Sockets page.

Changing the User Account (UNIX/Linux)

The Server Settings page allows you to change the user account for your web server on UNIX and Linux machines. All the server's processes run as this user.

You do not need to specify a server user if you chose a port number greater than 1024 and are not running as the `root` user (in this case, you do not need to be logged on as `root` to start the server). If you do not specify a user account here, the server runs with the user account you start it with. Make sure that when you start the server, you use the correct user account.

NOTE If you do not know how to create a new user on your system, contact your system administrator or consult your system documentation.

Even if you start the server as root, you should not run the server as root all the time. You want the server to have restricted access to your system resources and run as a non-privileged user. The user name you enter as the server user should already exist as a normal UNIX/Linux user account. After the server starts, it runs as this user.

If you want to avoid creating a new user account, you can choose the user nobody or an account used by another HTTP server running on the same host. On some systems, however, the user nobody can own files but not run programs.

To access the Server Settings page, perform the following steps:

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

Changing the Superuser Settings

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

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

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

1. Access the Administration Server and choose the Preferences tab.

2. Click the Superuser Access Control link.
3. Make the desired changes and click OK.

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

The superuser's user name and password are kept in a file called *server_root*/https-admserv/config/admpw. If you forget the user name, you can view this file to obtain the actual name; however, note that the password is encrypted and unreadable. The file has the format username:password. If you forget the password, you can edit the admpw file and simply delete the encrypted password. You can then go to the Server Manager forms and specify a new password.

CAUTION Because you can edit the admpw file, it is very important that you keep the server computer in a secure place and restrict access to its file system:

- On UNIX/Linux systems, consider changing the file ownership so that it's writable only by root or whatever system user runs the Administration Server daemon.
- On Windows systems, restrict the file ownership to the user account Administration Server uses.

Allowing Multiple Administrators

Multiple administrators can change specific parts of the server through distributed administration.

NOTE The default Directory Service must be an LDAP-based directory service for distributed administration to work.

With distributed administration you have two levels of users:

- **superuser** is the user listed in the file `server_root/https-admserv/config/admpw`. This is the user name (and password) you specified during installation. This user has full access to all forms in the Administration Server, except the Users & Groups forms, which depend on the superuser having a valid account in an LDAP server such as Sun ONE Directory Server.
- **administrators** go directly to the Server Manager forms for a specific server, including the Administration Server. The forms they see depend on the access control rules set up for them (usually done by the superuser). Administrators can perform limited administrative tasks and can make changes that affect other users, such as adding users or changing access control.

For an in-depth discussion on access control, see “What Is Access Control?,” on page 179 in Chapter 9, “Controlling Access to Your Server”.

NOTE Before you can enable distributed administration, you must install a Directory Server. For more information, see the Sun ONE Web Server *Installation and Migration Guide* and the Sun ONE Directory Server *Administrator’s Guide*.

To enable distributed administration, perform the following steps:

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

   To create a group, perform the following steps:

   a. Choose the Users & Groups tab.

   b. Click the New Group link.

   c. Create an “administrators” group in the LDAP directory and add the names of the users you want to have permission to configure the Administration Server, or any of the servers installed in its server root. All users in the “administrators” group have full access to the Administration Server, but you can use access control to limit the servers and forms they will be allowed to configure.

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

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

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

Specifying Log File Options

The Administration Server log files record data about the server, including the types of errors encountered and information about server access. Viewing these logs allows you to monitor server activity and troubleshoot problems by providing data like the type of error encountered and the time certain files were accessed.

You can specify the type and format of the data recorded in the Administration Server logs using the Log Preferences page. For instance, you can choose to log data about every client who accesses the Administration Server or you can omit certain clients from the log. In addition, 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.

Access the Administration Server Log Preferences page by choosing the Preferences tab, then clicking the Logging Options link.

For more information, see the Logging Options page in the online help, and Chapter 10, “Using Log Files”.

Viewing Log Files

The Administration Server log files are located in `admin/logs` in your server root directory. For example, on Windows, the path to your log files might look like `c:\Sun\server6\https-admserv\logs`. You can view both the error log and the access log through the Sun ONE Web Server console or using a text editor.

The Access Log File

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

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

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

For more information, see the View Error Log page in the online help, and Chapter 10, “Using Log Files”.

The Error Log File

The error log lists all the errors the server has encountered since the log file was created. It also contains informational messages about the server, such as when the server was started and who tried unsuccessfully to log in to the server.

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

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

For more information, see the View Access Log page in the online help, and Chapter 10, “Using Log Files”.

Archiving Log Files

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

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

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

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

Using schedulerd Control-based Log Rotation (UNIX/Linux)

You can configure several features of your Sun ONE Web Server to operate automatically and set to begin at specific times. The `schedulerd` control daemon checks the computer clock and then spawns processes at certain times. (These settings are stored in the `schedulerd` file.)

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

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

To restart, start, or stop the `schedulerd` control daemon, perform the following steps:

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

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

Configuring Directory Services

You can store and manage information such as the names and passwords of your users in a single Directory Server using an open-systems server protocol called the Lightweight Directory Access Protocol (LDAP). You can also configure the server to allow your users to retrieve directory information from multiple, easily accessible network locations.

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

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

3. Make the desired changes and click OK.

For more information, see the Configure Directory Service page in the online help.

Restricting Server Access

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

You can set access control globally for all servers through the Administration Server or for a resource within a specific server instance through the Server Manager. For more information about setting access control for a resource, see “Setting Access Control,” on page 192 in Chapter 9, “Controlling Access to Your Server”.

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

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

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

   The Administration Server displays the access control rules for the server you specified.
4. Make the desired access control changes and click OK. For more information, see the Restrict Access page in the online help.

Chapter 6

Using Certificates and Keys

This chapter describes the use of certificates and keys authentication to secure the Sun ONE Web Server 6.1. It describes how to activate the various security features designed to safeguard your data, deny intruders access, and allow access to those you want. Sun ONE Web Server 6.1 incorporates the security architecture of all Sun ONE servers: it's built on industry standards and public protocols for maximum interoperability and consistency.

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

The process of securing your web server is explained in detail in the following sections:

Certificate-based Authentication

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

Using Certificates for 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.

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

In addition to a public key and the name of the entity identified by the certificate, a certificate also includes 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*.

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

Server Authentication

Server authentication refers to the confident identification of a server by a client; that is, identification of the organization assumed to be responsible for the server at a particular network address.

Client Authentication

Client authentication refers to the confident identification of a client by a server; that is, identification of the person assumed to be using the client software. Clients can have multiple certificates, much like a person might have several different pieces of identification.

Virtual Server Certificates

You can have a different certificate database per virtual server. Each virtual server database can contain multiple certificates. Virtual servers can also have different certificates within each instance.

Creating a Trust Database

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

When you create the trust database, you specify a password that will be used for a key-pair file. You will also need this password to start a server using encrypted communications. For a list of guidelines to consider when changing a password, see “Changing Passwords or PINs” on page 150.

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 will 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/<serverid-hostname>-key3.db.

The Administration Server can only have one trust database. Each server instance can have its own trust database. Virtual servers are covered by the trust database created for their server instance.

Creating a Trust Database

To create a trust database, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click on the Create Database link.
3. Enter a password for the database.
4. Repeat.
5. Click OK.

6. For the Server Manager, click Apply, and then Restart for changes to take effect.

Using password.conf

By default, the web server prompts the administrator for the key database password before starting up. If you want to be able to restart an unattended web server, you need to save the password in a `password.conf` file. Only do this if your system is adequately protected so that this file and the key databases are not compromised.

Normally, you cannot start an 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 plain text 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 them.

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 its access, even if you do not use the file. The directory should have read / write permissions for the administration server user and the web server user. Protecting the directory prevents others from creating a false `password.conf` file. You cannot protect directories or files on FAT file systems by restricting access to them.

Start an SSL-enabled Server Automatically

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

1. Make sure SSL is on.
2. Create a new `password.conf` file in the `config` subdirectory of the server instance.
   - If you are using the internal PKCS#11 software encryption module that comes with the 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 with the password. For example:

  `nFast:`*`your_password`*

3. Stop and restart your server for the new setting to take effect.

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

Requesting and Installing a VeriSign Certificate

VeriSign is Sun ONE Web Server's preferred certificate authority. VeriSign's VICE protocol simplifies the certificate request process. VeriSign has the advantage of being able to return their certificate directly to your server.

After creating a certificate trust database for your server, you can request a certificate and submit it to a Certificate Authority (CA). If your company has its own internal CA, request your certificate from them. If you plan to purchase your certificate from a commercial CA, choose a CA and ask for the specific format of the information they require. A list of available certificate authorities including links to their sites, is available on the Request a Certificate page. For more information on what CAs may require, a list of Certificate Authorities is available through both Server Administrator, and Server Manager Security Pages under Request a Certificate.

The Administration Server can have only one server certificate. Each server instance can have its own server certificate. You can select a server instance certificate for each virtual server.

Requesting a VeriSign Certificate

To request a VeriSign Certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Request VeriSign Certificate link.
3. Review the steps required.

4. Click OK.
5. Follow the VeriSign procedure.

Installing a VeriSign Certificate

If you request and receive approval for a VeriSign certificate, it should appear in the drop-down list of the Install VeriSign Certificate page in one to three days. To install a VeriSign Certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.
2. Click the Install VeriSign Certificate link.
3. Choose internal (software) from the drop-down list for cryptographic module, unless you will use an external encryption module.
4. Enter your Key Pair File Password or PIN.
5. Select the Transaction ID to Retrieve from the drop-down list.

   You will usually want the last one.
6. Click OK.
7. For the Server Manager, click Apply, and then Restart for changes to take effect.

Requesting and Installing Other Server Certificates

Besides VeriSign, you can request and install certificates from other certificate authorities. A list of CAs is available through both Server Administrator, and Server Manager Security Pages under Request a Certificate. Your company or organization may provide its own internal certificates. This section describes how you would request and install these other types of server certificates.

Required CA Information

Before you begin the request process, make sure you know what information your CA requires. Whether you are requesting a server certificate from a commercial CA or an internal CA, you need to provide the following information:

- **Common Name** must be the fully qualified hostname used in DNS lookups (for example, *www.sun.com*). This is the hostname in the URL that a browser uses to connect to your site. If these two names don't match, a client is notified that the certificate name doesn't match the site name, creating doubt about the authenticity of your certificate. Some CAs might have different requirements, so it's important to check with them.

  You can also enter wildcard and regular expressions in this field if you are requesting a certificate from an internal CA. Most vendors would not approve a certificate request with a wildcard or regular expression entered for common name.

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

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

- **Organizational Unit** is an optional field that describes an organization within your company. This can also be used to note a less formal company name (without the *Inc.*, *Corp.*, and so on).

- **Locality** is an optional field that usually describes the city, principality, or country for the organization.

- **State or Province** is usually required, but can be optional for some CAs. Note that most CAs won't accept abbreviations, but check with them to be sure.

- **Country** is a required, two-character abbreviation of your country name (in ISO format). The country code for the United States is US.

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

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

Some commercial CAs offer certificates 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.sun.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 a certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Request a Certificate link.

3. Select if this is a new certificate or a certificate renewal.

   Many certificates expire after a set period of time, such as six months or a year. Some CAs will automatically send you a renewal.

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

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

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

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

6. Enter the password for your key-pair file.

   This is the password you specified when you created the trust database, unless you selected a cryptographic module other than the internal module. The server uses the password to get your private key and encrypt a message to the CA. The server then sends both your *public key* and the encrypted message to the CA. The CA uses the public key to decrypt your message.

7. Enter your identification information.

   The format of this information varies by CA. For a general description of these fields, alist of Certificate Authorities is available through both Server Administrator, and Server Manager Security Pages under Request a Certificate. Note that most of this information usually isn't required for a certificate renewal.

8. Double-check your work to ensure accuracy.

   The more accurate the information, the faster your certificate is likely to be approved. If your request is going to a certificate server, you'll be prompted to verify the form information before the request is submitted.

9. Click OK.

10. For the Server Manager, click Apply, and then Restart for changes to take effect.

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

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

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

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

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

Installing Other Server Certificates

When you receive your certificate back from the CA, it will be 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 (CA) and is used to sign certificates issued by that authority. A CA certificate can in turn be signed by the CA certificate of a parent CA, and so on, up to a root CA.

NOTE If your CA doesn't automatically send you their certificate, you should request it. Many CAs include their certificate in the email with your certificate, and your server installs both certificates at the same time.

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

Installing a Certificate

To install a certificate, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Install Certificate link.
3. Check the type of certificate you are installing:
   - This Server is for a single certificate associated only with your server.

- Server Certificate Chain is for a CA's certificate to include in a certificate chain.
- Trusted Certificate Authority (CA) is for a certificate of a CA that you want to accept as a trusted CA for client authentication.

4. Select the Cryptographic Module from the drop-down list.
5. Enter the Key-Pair File Password.
6. Leave the a name for the certificate field blank if it will be the only one used for this server instance, unless:

   - Multiple certificates will be used for virtual servers

     Enter a certificate name unique within the server instance

   - Cryptographic modules other than internal are used

     Enter a certificate name unique across all server instances within a single cryptographic module

   If a name is entered, it will be displayed in the Manage Certificates list, and should be descriptive. For example, "United States Postal Service CA" is the name of a CA, and "VeriSign Class 2 Primary CA" describes both a CA and the type of certificate. When no certificate name is entered, the default value is applied.

7. Select either:

   - Message is in this file and enter the full pathname to the saved email
   - Message text (with headers) and paste the email text

     If you copy and paste the text, be sure to include the headers "Begin Certificate" and "End Certificate"—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.

10. For the Server Manager, click Apply, and then Restart for changes to take effect.

The certificate is stored in the server's certificate database. The filename will be `<alias>-cert8.db`. For example:

`https-`*`serverid-hostname`*`-cert8.db`

Migrating Certificates When You Upgrade

If you are migrating from iPlanet Web Server 4.1 or 6.0, your files, including your trust and certificate databases, will be updated automatically.

If you are upgrading from an Enterprise Server 3.x, you will need to migrate your trust and certificate databases. Make sure that Sun ONE Web Server 6.1 Administration Server user has read and write 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 your server has security enabled. You can also migrate keys and certificates by themselves using the Security tabs in the Administration Server page and the Server Manager page.

In previous versions, a certificate and key-pair file was referred to by an alias which could be used by multiple server instances. The Administration Server managed all the aliases and their constituent certificates. In Sun ONE Web Server 6.1, 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.

You manage the trust database and its constituent certificates, including the server certificate and all the included Certificate Authorities, from the Administration Server for its self, 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 the Certificate Authorities listed in your previous database are migrated to the Sun ONE Web Server 6.1 database. If duplicate CAs occur, use the previous CA until it expires. Do not attempt to delete duplicate CAs.

Using the Built-in Root Certificate Module

The dynamically loadable root certificate module included with Sun ONE Web Server 6.1 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 than before. In the past, you were required to delete the old root

certificates one at a time, 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 Sun ONE Web Server, or in Service Packs.

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)
- `libnssckbi.sl` (on HP-UX)
- `nssckbi.dll` (on Windows)

If you later wish to restore the root certificate module, you can copy the extension from `bin/https/lib` (UNIX and HP) or `bin\https\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 certificate lists, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Manage Certificates link.

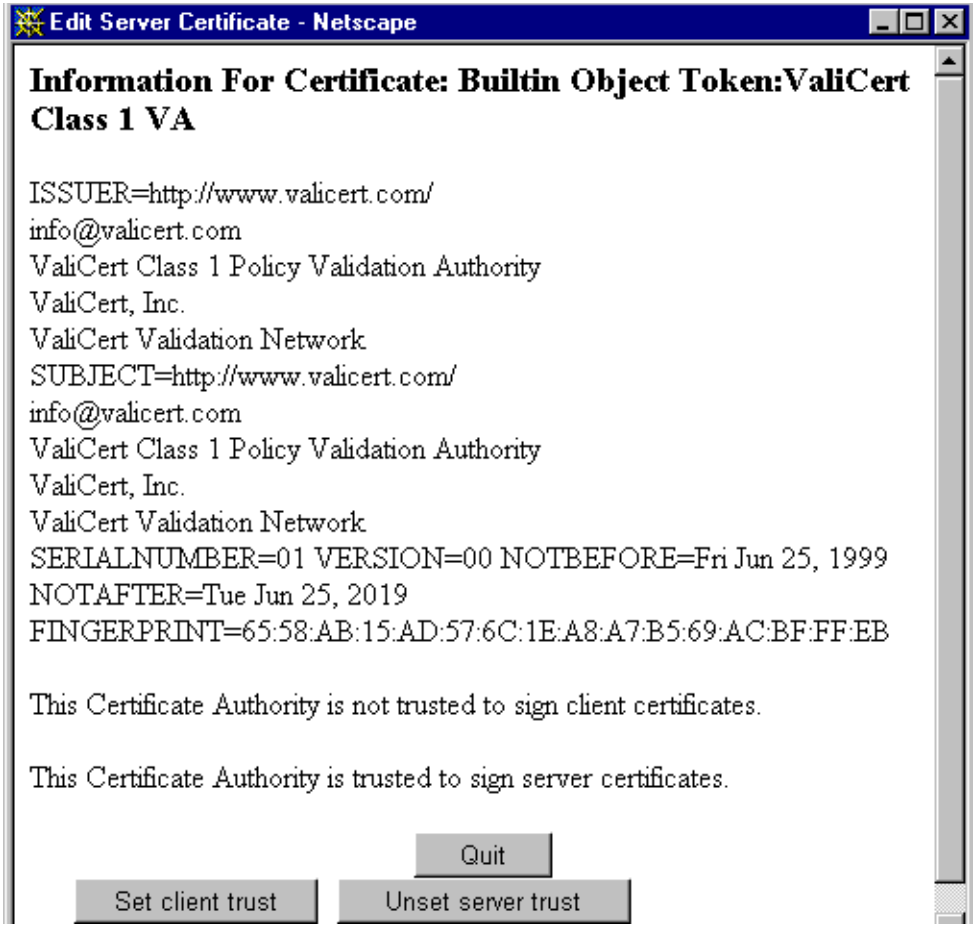
   - If you are managing a certificate for a default configuration using the internal cryptographic module, a list of all installed certificates with their type and expiration date is displayed. All certificates are stored in the directory *`server_root`*`/alias`.

- If you are using an external cryptographic module, such as a hardware accelerator, you will first need to enter your password for each specific module and click OK. The certificate list will update to include certificates in the module.

3. Click the Certificate Name you wish to manage.

   An Edit Server Certificate page appears with management options for that type of certificate. Only CA certificates will allow you to set or unset client trust. Some external cryptographic modules will not allow certificates to be deleted.

   Edit Server Certificate



4. In the Edit Server Certificate window you may select:

- Delete Certificate or Quit for certificates obtained internally
- Set client trust, Unset server trust, or Quit for CA certificates

5. Click OK.
6. For the Server Manager, click Apply, and then Restart for changes to take effect.

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, for example, a user changes offices or leaves the organization before the certificate expires, the certificate is revoked, and its data appears in a CRL. If a key is tampered with or otherwise compromised, the key and its data appear in a CKL. Both CRLs and CKLs are produced and periodically updated by a CA.

Installing a CRL or CKL

To obtain a CRL or CKL from a CA, perform the following steps:

1. Obtain the CA’s URL for downloading CRLs or CKLs.
2. Enter the URL in your browser to access the site.
3. Follow the CA's instructions for downloading the CRL or CKL to a local directory.
4. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.
5. Click the Install CRL / CKLs link.
6. Select either:

   - Certificate Revocation List

- Compromised Key List

7. Enter the full path name to the associated file.
8. Click OK.
   - If you selected Certificate Revocation List, the Add Certificate Revocation List page will appear listing CRL information.
   - If you selected Compromised Key List, the Add Compromised Key List page will appear listing CKL information.

NOTE If a CRL or CKL list already exists in the database, a Replace Certificate Revocation List or Replace Compromised Key List page will appear.

9. Click Add.
10. Click OK.
11. For the Server Manager, click Apply, and then Restart for changes to take effect.

Managing CRLs and CKLs

To manage CRLs and CKLs, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.
2. Click the Manage CRL/CKLs link.

   The Manage Certificate Revocation Lists /Compromised Key Lists page appears with all installed Server CRLs and CKLs listed along with their expiration dates.
3. Select a Certificate Name from either the Server CRLs or Server CKLs list.
4. Choose:
   - Delete CRL
   - Delete CKL

5. For the Server Manager, click Apply, and then Restart for changes to take effect.

Setting Security Preferences

Once you have a certificate, you can begin securing your server. Several security elements are provided by Sun ONE Web Server.

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. Sun ONE Web Server 6.1 includes supports SSL and 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 need to 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 SSL2, SSL3, and TLS protocols.

NOTE Improvements to security and performance were made after SSL version 2.0; you should not use SSL 2 unless you have clients that are not capable of using SSL 3. Client certificates are not guaranteed to work with SSL 2 ciphers.

The encryption process alone isn't 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 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, check them in the list. Unless you have a compelling reason not to use a specific cipher, you should check them all. However, you may not wish to enabling ciphers with less than optimal encryption.

CAUTION Do not select "No Encryption, only MD5 message authentication". If no other ciphers are available on the client side, the server will default to this setting and no encryption will occur.

SSL and TLS Protocols

Sun ONE Web Server 6.1 supports the Secure Sockets Layer (SSL) and the Transport Layer Security (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, perform the following steps:

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Configure Directory Service link.
3. Select Yes to use Secure Sockets Layer (SSL) for connections.
4. Click Save Changes.
5. Click OK to change your port to the standard port for LDAP over SSL.

Enabling Security for Listen Sockets

You can secure your server's listen sockets by:

- 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 when you edit an existing listen socket.

Turning Security On When Creating a Listen Socket

To turn security on when creating a new listen socket, perform the following steps:

1. Access the Server Manager and select the server instance the listen socket will be created in from the drop-down list.
2. Select the Preferences tab, if not already displayed.
3. Choose the Edit Listen Sockets link.

   The Edit Listen Sockets page is displayed.
4. Click the New button.

   The Add Listen Socket page is displayed.
5. Enter the required information and select a default virtual server.
6. To turn security on, select Enabled from the Security drop-down list.
7. Click OK
8. Click Apply, and then Restart for changes to take effect.

NOTE You will need to use the Edit Listen Sockets link to configure the security settings after a listen socket is created.

Turning Security On When Editing a Listen Socket

You can also turn security on when editing a listen socket from either the Administration Server or the Server Manager. To turn security on when editing a listen socket, perform the following steps:

1. Access either the Administration Server or the Server Manager and choose the Security tab.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Select the Preferences tab, if not already displayed.

3. Choose the Edit Listen Sockets link.

   The Edit Listen Sockets page is displayed.

4. To edit a listen socket, click the Listen Socket ID of the listen socket you want to edit.

   The Edit Listen Socket page is displayed.

5. To turn security on for the listen socket, select Enabled from the Security drop-down list.

6. Click OK.

7. For the Server Manager, click Apply, and then Restart for changes to take effect.

Selecting a Server Certificate for a Listen Socket

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

NOTE You must have at least one certificate is installed.

To select a server certificate for your listen socket to use, perform the following steps:

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

   For the Server Manager you must first select the server instance from the drop-down list.

2. Choose the Edit Listen Sockets link.

   The Edit Listen Sockets page is displayed.

3. To edit a listen socket, click the Listen Socket ID of the listen socket you want to edit.

   The Edit Listen Socket page is displayed.

4. To turn security on for the listen socket, select Enabled from the Security drop-down list.

   .

NOTE If you have an external module installed, the Manage Server Certificates page will appear requiring the external module's password before you can continue.

5. Select a server certificate from the drop-down Server Certificate Name list for the listen socket.

   The list contains all internal and external certificates installed.

NOTE If no server certificates are installed, a warning message to this effect is displayed in place of the Server Certificate Name drop-down list.

6. Click OK
7. For the Server Manager, click Apply, and then Restart for changes to take effect.

Selecting Ciphers

To protect the security of your web server, you should enable SSL. You can enable the SSL 2.0, SSL 3.0, and TLS encryption protocols and select the various cipher suites. SSL and TLS can be enabled on the listen socket for the Administration Server. Enabling SSL and TLS on a listen socket for the Server Manager will set those security preferences for all virtual servers associated with that listen socket.

If you wish to have unsecured virtual servers, they must all be configured to the same listen socket with security turned off.

The default settings allow the most commonly used ciphers. Unless you have a compelling reason why you don't want to use a specific cipher suite, you should allow them all. For more information regarding specific ciphers, see *Introduction to SSL*.

NOTE You must have at least one certificate installed.

The default and recommended setting of the `tlsrollback` parameter is `true`. This configures the server to detect man-in-the-middle version rollback attack attempts. Setting this value to `false` may be required for interoperability with some clients that incorrectly implement the TLS specification.

Note that setting `tlsrollback` to `false` leaves connections vulnerable to version rollback attacks. Version rollback attacks are a mechanism by which a 3rd party can force a client and server to communicate using an older, less secure protocol such as SSLv2. Because there are known deficiencies in the SSLv2 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, perform the following steps:

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

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Edit Listen Sockets link.

   The Edit Listen Sockets page appears. For a secure listen socket, the Edit Listen Socket page displays the available cipher settings.

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".

3. Check the checkboxes corresponding to the required encryption settings.

NOTE Check both TLS and SSL3 for Netscape Navigator 6.0. For TLS Rollback also check TLS, and make sure both SSL3 and SSL2 are disabled.

4. Click OK.

5. For the Server Manager, click Apply, and then Restart for changes to take effect.

NOTE When you apply changes after turning on security for a listen socket, the `magnus.conf` file is automatically modified to show security on, and all virtual servers associated with the listen socket are automatically assigned the default security parameters.

Once you have enabled SSL 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.sun.com:443`.

If you use the default secure http port number (443), you don't have 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. Security must be set to 'on' for virtual server security settings to work. SSL properties for virtual servers can be found on a per-server basis in the `SSLPARAMS` element of the `server.xml` file.

To set values for your SSL configuration file directives, perform the following steps:

1. Access the Server Manager and select the server instance of the virtual server from the drop-down list.
2. Ensure that security is enabled for the listen socket you want to configure. To do so, perform the following steps:
   a. Click the Edit Listen Sockets link.
   b. Click the Listen Socket ID corresponding to the listen socket you want to enable security on.

      This takes you to the Edit Listen Socket page.
   c. Select Enabled from the Security drop-down list.
   d. Click OK.
3. Click the Magnus Editor link.
4. Select SSL Settings from the drop-down list and click Manage.
5. Enter the values for:
   - SSLSessionTimeout
   - SSLCacheEntries
   - SSL3SessionTimeout

6. Click OK
7. Click Apply, and then Restart for changes to take effect.

These SSL Configuration File Directives are described below:

SSLSessionTimeout

The `SSLSessionTimeout` directive controls SSL2 session caching.

Syntax

`SSLSessionTimeout` seconds

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 SSL3 and TLS session caching.

Syntax

`SSL3SessionTimeout` seconds

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

Using External Encryption Modules

Sun ONE Web Server 6.1 supports the following methods of using external cryptographic modules such as smart cards or token rings:

- PKCS#11
- FIPS-140

You will need to add the PKCS #11 module before activating the FIPS-140 encryption standard.

Installing the PKCS#11Module

Sun ONE Web 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.

Using modutil to Install a PKCS#11 Module

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

To install the PKCS#11 module using `modutil`, perform the following steps:

1. Make sure all servers, including the Administration server, are turned off.
2. Go to the `server_root/alias directory` containing the databases.
3. Add `server_root/bin/https/admin/bin` to your PATH.
4. Locate `modutil` in `server_root/bin/https/admin/bin`.
5. Set the environment. For example:
   - On UNIX: `setenv`

     `LD_LIBRARY_PATH server_root/bin/https/lib:${LD_LIBRARY_PATH}`
   - On IBM-AIX: `LIBPATH`
   - On HP-UX: `SHLIB_PATH`
   - On Windows, add it to the `PATH`

     `LD_LIBRARY_PATH server_root/bin/https/bin`

     You can find the PATH for your machine listed under: `server_root/https-admin/start`.
6. Enter the command: `modutil`.

   The options will be listed.
7. Perform the actions required.

   For example, to add the PCKS#11 module in UNIX you would enter:

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

Using pk12util

The `pk12util` allows you to export certificates and keys from your internal database and to 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, perform the following steps:

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

     `LD_LIBRARY_PATH/server_root/bin/https/lib:${LD_LIBRARY_PATH}`
   - On IBM-AIX: `LIBPATH`
   - On HP-UX: `SHLIB_PATH`
   - On Windows, add it to the `PATH`

     `LD_LIBRARY_PATH server_root/bin/https/bin`

     You can find the PATH for your machine listed under: `server_root/https-admin/start`.
5. Enter the command: `pk12util`.

   The options will be listed.
6. Perform the actions required.

   For example, in UNIX you would enter:

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

Importing with pk12util

To import a certificate and key into an internal or external PKCS#11 module, perform the following steps:

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

     `LD_LIBRARY_PATH/server_root/bin/https/lib:${LD_LIBRARY_PATH}`
   - On IBM-AIX: `LIBPATH`
   - On HP-UX: `SHLIB_PATH`
   - On Windows, add it to the `PATH`

     `LD_LIBRARY_PATH server_root/bin/https/bin`

     You can find the PATH for your machine listed under: `server_root/https-admin/start`.
5. Enter the command: `pk12util`.

   The options will be listed.
6. Perform the actions required.

   For example, in UNIX you would 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 quote marks.
7. Enter the database password.
8. Enter `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`, 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’ll need to specify the certificate name for the listen socket it runs on.

Selecting the Certificate Name for a Listen Socket

To select the certificate name for the listen socket, perform the following steps:

NOTE 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 it. For more information, see “Enabling Security for Listen Sockets”.

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

   For the Server Manager you must first select the server instance from the drop-down list.
2. Select the Preferences tab, if not already selected.
3. Click the Edit Listen Sockets link.

   The Edit Listen Sockets page appears.
4. Click the Listen Socket Id link corresponding to the listen socket you want to associate with a certificate.

   The Edit Listen Socket page appears.
5. Select a server certificate from the drop-down Server Certificate Name list for the listen socket.

   The list contains all internal and external certificates installed.

NOTE If no server certificates are installed, a warning to this effect is displayed in place of the Server Certificate Name drop-down list.

6. Click OK

7. For the Server Manager, click Apply, and then Restart for changes to take effect.

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 Security tab Create Database page to set the password.

FIPS-140 Standard

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

To enable FIPS-140, perform the following steps:

1. Install the plug-in following the FIPS-140 instructions.
2. Access either the Administration Server or the Server Manager and choose the Preferences tab.

   For the Server Manager you must first select the server instance from the drop-down list.

3. Click the Edit Listen Sockets link.

   The Edit Listen Sockets page appears. For a secure listen socket, the Edit Listen Socket 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".

4. Select Enabled from the SSL Version 3 drop-down list, if it is not already selected.
5. Check the appropriate FIPS-140 cipher suite:
   - (FIPS) DES with 56 bit encryption and SHA message authentication
   - (FIPS) Triple DES with 168 bit encryption and SHA message authentication
6. Click OK.
7. For the Server Manager, click Apply, and then Restart for changes to take effect.

Setting Client Security Requirements

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

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 will send a response to a query.

Sun ONE Web 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 under Security in the Administration Server. There are four types of CAs:

- Untrusted CA (will not be matched)
- Trusted Server CA (will not be matched)

- Trusted Client CA (will be matched)
- Trusted Client/Server CA (will be matched)

You can configure the web server to refuse any client that doesn't have a client certificate from a trusted CA. To accept or reject trusted CAs, you must have set client trust for the CA. For more information, see "Managing Certificates" on page 121.

Sun ONE Web Server will log an error, reject the certificate, and return a message to the client if the certificate has expired. You can also view which certificates have expired in the Administration Servers 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, see "Mapping Client Certificates to LDAP" on page 140.

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 188.

You can also process information from client certificates. For more information, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide*.

To Require Client Authentication

To require client authentication, perform the following steps:

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

   For the Server Manager you must first select the server instance from the drop-down list.

2. Click the Edit Listen Sockets link.

   The Edit Listen Sockets page appears.

3. Click the Listen Socket Id link corresponding to the listen socket you are requiring client authentication for.

   The Edit Listen Socket page appears.

4. To require client authenticate for the listen socket, select Required from the Client Authentication drop-down list.

5. Click OK.
6. For the Server Manager, click Apply, and then Restart for changes to take effect.

NOTE Currently, there is a single certificate trust database per web server instance. All the secure virtual servers running under that server instance share the same list of trusted client CAs. If two virtual servers require different trusted CAs, then these virtual servers should be run in different server instances with separate trust databases.

Mapping Client Certificates to LDAP

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

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

NOTE Before mapping client certificates to LDAP, you also need to set up the required ACLs; for more information, see Chapter 9, "Controlling Access to Your Server".

The server tries to match the CA to the list of trusted CAs in the Administration Server. If there isn't a match, Sun ONE Web 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:

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

Once the server knows where to start its search and what it needs to search for (step 1), it performs the search in the LDAP directory (step 2). If it finds no matching entry or more than one matching entry, and the mapping is *not* set to verify the certificate, the search fails. For a complete list of the expected search result behavior, see the following Table 5-1 table. Note that you can specify the expected behavior in the ACL; for example, you can specify that Sun ONE Web Server accepts only you if the certificate match fails. For more information regarding how to set the ACL preferences, see "Using Access Control Files" on page 188.

Table 6-1 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 userid, email, 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 its search

- What certificate attributes the server should use as search criteria when searching for the entry in the LDAP directory
- Whether or not the server goes through an additional verification process

The certificate mapping file is located in the following location:

server_root/userdb/certmap.conf

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

```
certmap <name> <issuerDN>

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

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

```
certmap 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 ONE Directory Server and experiencing problems in matching the `issuerDN`, 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 you set `DNComps` to use the `o` and `c` attributes of the DN, the server starts the search from the `o=<org>, c=<country>` entry in the LDAP directory, where `<org>` and `<country>` are replaced with values from the DN in the certificate.

  Note the following situations:

- If there isn't a `DNComps` entry in the mapping, the server uses either the `CmapLdapAttr` setting or the entire subject DN in the client certificate (that is, the end-user's information).
- If the `DNComps` entry is present but has no value, the server searches the entire LDAP tree for entries matching the filter.

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

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

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

Table 6-2 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`	Email address

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

- `verifycert` tells the server whether it should compare the client's certificate with the certificate found in the LDAP directory. It takes two values: on, and off. You should only use this property if your LDAP directory contains certificates. This feature is useful to ensure your end-users have a valid, unrevoked certificate.
- `CmapLdapAttr` is a name for the attribute in the LDAP directory that contains subject DNs from all certificates belonging to the user. The default for this property is `certSubjectDN`. This attribute isn't a standard LDAP attribute, so to use this property, you have to extend the LDAP schema. For more information, see *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 the search doesn't find any entries, the server retries the search using the `DNComps` and `FilterComps` mappings.

  This approach to matching a certificate to an LDAP entry is useful when it's difficult to match entries using `DNComps` and `FilterComps`.
- `Library` is a property whose value is a pathname to a shared library or DLL. You only need to use this property if you create your own properties using the certificate API. For more information, see the *NSAPI Programmer's Guide.*
- `InitFn` is a property whose value is the name of an init function from a custom library. You only need to use this property if you create your own properties using the certificate API.

For more information on these properties, refer to the examples described in "Sample Mappings" on page 145.

Creating Custom Properties

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

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 you can use the `certmap.conf` file.

Example #1

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

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

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

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

Example #2

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

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` won't match because there isn't a space between the `o` and the `c` attributes.

Example #3

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

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

If the client certificate subject is:

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

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

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

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

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

Setting Stronger Ciphers

The Stronger Ciphers option 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, Sun ONE Web 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, Sun ONE Web 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 now 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 (not a URI) 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 will occur the next time the same client connects to the server.

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

To Set Stronger Ciphers, perform the following steps:

1. Access the Server Manager and select the server instance from the drop-down list.
2. Click the Virtual Server Class tab.
3. Select a class from the drop-down list and click Manage.

   The Class Manager page appears.
4. Choose the Content Mgmt tab.

5. Select Stronger Ciphers.
6. Choose to edit:
   - from the drop down list
   - by clicking Browse
   - by clicking Wildcard
7. Select the secret key size restriction:
   - 168 bit or larger
   - 128 bit or larger
   - 56 bit or larger
   - No restrictions
8. Enter the file location of the message to reject access.
9. Click OK.
10. Click Apply.
11. Select hard start / restart or dynamically apply

For more information, see *Introduction to SSL*.

Considering Additional Security Issues

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

So in addition to enabling encryption on your server, you should take extra security precautions. For example, put the server machine into a secure room, and don't allow individuals you don't trust to upload programs to your server.

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

- Limit Physical Access
- Limit Administration Access
- Choosing Solid Passwords
- Changing Passwords or PINs

- Limiting Other Applications on the Server
- Preventing Clients from Caching SSL Files
- Limiting Ports
- Knowing Your Server's Limits
- Making Additional Changes to Protect Servers

Limit Physical Access

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

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

Limit Administration Access

If you use remote configuration, be sure to 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 regarding clusters, see "About Clusters" on page 157.

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

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

Creating Hard-to-Crack Passwords

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

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

- Passwords should be 6-14 characters long. (Mac passwords cannot be longer than 8 characters)
- Do not use the "illegal" characters: *, ", or spaces
- Do not use dictionary words (any language)
- Do not make common letter substitutions, like replacing 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's 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 machine. For a list of guidelines to consider when changing a password, see “Creating Hard-to-Crack Passwords” on page 150.

Changing Passwords

To change your trust database/key-pair file password for the Administration Server or an server instance, perform the following steps:

1. Access either the Administration Server or the Server Manager.

   For the Server Manager you must first select the server instance from the drop-down list.

2. Select the Change Password link.

3. Select the security token on which you want to change the password from the drop-down list.

   By default this is ‘internal’ for the internal key database. If you have PKCS#11 modules installed, you will see all the tokens listed. Click the Change Password link.

4. Enter your current password.

5. Enter your new password

6. Enter it again.

7. Click OK.

8. For the Server Manager, click Apply, and then Restart for changes to take effect

Make sure your key-pair file is protected. The Administration Server stores key-pair files in the directory *server_root*`/alias`. Consider making the files and directory readable only to Sun ONE servers installed on your computer.

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

Limiting Other Applications on the Server

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

UNIX and Linux

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

Windows

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

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

Preventing Clients from Caching SSL Files

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

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

Limiting Ports

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

Knowing Your Server's Limits

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

Being aware of these limitations helps you 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 machine? What happens to those numbers after the SSL connection is terminated? You should be responsible for securing any information clients send to you through SSL.

Making Additional Changes to Protect Servers

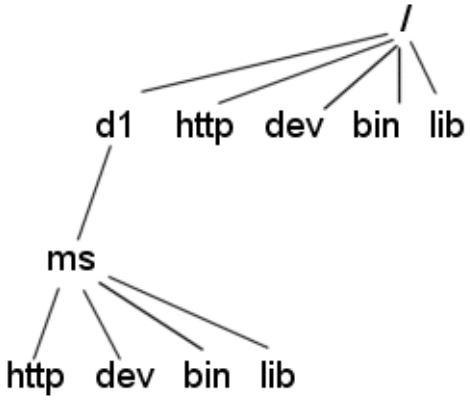
If you want to have both protected and unprotected servers, you should operate the unprotected server on a different machine from the protected one. If your resources are limited and you must run an unprotected server on the same machine as your protected server, do the following.

- Assign proper port numbers. Make sure that the protected server and the unprotected server are assigned different port numbers. The registered default port numbers are:
  - 443 for the protected server
  - 80 for the unprotected server
- For UNIX or Linux, enable the `chroot` feature for the document root directory. The unprotected server should have references to its document root redirected using `chroot`.

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

However, if you use `chroot`, you need to set up the full directory structure required by Sun ONE Web Server under the alternative root directory, as shown in the following illustration:

Example of chroot Directory Structure



Specifying chroot for a Virtual Server Class

You can specify the `chroot` directory for a virtual server class by performing the following steps:

1. Access the Server Manager and select the server instance from the drop-down list.
2. Select the Virtual Server Class tab.
3. Click the Edit Classes link.
4. Make sure the Option is set to Edit for the class in which you wish to specify `chroot`.
5. Click the Advanced button for that class.

   The Virtual Servers CGI Settings page appears.
6. Enter the full pathname in the Chroot field.
7. Click OK.
8. Click Apply.
9. Choose Load Configuration Files to dynamically apply.

Specifying chroot for a Virtual Server

You can specify the `chroot` directory for a specific virtual server by performing the following steps:

1. Access the Server Manager and select the server instance from the drop-down list.
2. Select the Virtual Server Class tab.
3. Click on the link for the virtual server you wish to specify the `chroot` directory for from the Tree View of the Server.
4. Select the Settings tab.

   The Settings page appears.
5. Enter the full pathname in the Set to field next to Chroot Directory.
6. Click OK.
7. Click Apply.
8. Choose Load Configuration Files to dynamically apply.

You can also specify the `chroot` directory for a virtual server using the Class Manager Virtual Servers tab and the CGI Settings link.

For more information regarding how to specify a `chroot` directory for a virtual server, see the Sun ONE Web Server 6.1 *Programmer's Guide.*

Chapter 7

Managing Server Clusters

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

This chapter includes the following sections:

About Clusters

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

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

- Create a central place for administering all Sun ONE Web Servers

- Share one or more configuration files between servers
- Start and stop all servers from one “master” Administration Server
- View the access and error logs for the servers you selected

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

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

Guidelines for Using Server Clusters

When you configure a cluster, the master Administration Server containing the information about all clusters communicates with each individual cluster’s Administration Server. The administration server for each cluster must be given the same administration user name and password that the master Administration Server will have.

Before you can create a cluster, all of the servers you want to include in the cluster must be installed. For example, if you want three clusters of five Sun ONE Web Servers per cluster, you would need to:

1. Install all of the servers on the computers where they’ll run using the same administration user name and password as the master Administration Server.
2. Configure one of the Sun ONE Web Servers in each cluster as the Administration Server.
3. Configure one single cluster’s administration server as the master Administration Server for all clusters. It doesn’t matter which server you choose as the master administration server.

CAUTION Clusters can only be homogeneous. All servers in the cluster must be either UNIX or Windows. Combining UNIX and Windows servers in the same cluster may cause the server to hang or crash.

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

- Install all of the servers you want to include in a particular cluster prior to creating any clusters.
- Make sure all servers in a cluster must be version 6.1 Sun ONE Web Servers.
- Make sure all cluster-specific Administration Servers have the same userid and password as the master administration server. You can use distributed administration to set up multiple administrators on each Administration Server.
- Install servers on any computer in a network, as long as all computers in the cluster are Windows or UNIX.
- You can designate any cluster-specific Administration Server as the master administration server.
- Make sure the master Administration Server has access to each cluster-specific Administration Server. The master Administration Server retrieves information about all installed Sun ONE Web Servers.
- Make sure all Administration Servers are Sun ONE Web Server version 6.0 or 6.1 and use the same protocol, HTTP or HTTPS. Only Sun ONE Web Server 6.0 or 6.1 servers are supported for addition to clusters.
- If you change the protocol of one Administration Server in a cluster, you must change the protocols for all Administration Servers. Then use the Modify Server interface to modify the individual servers in the cluster.

Setting Up a Cluster

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

1. Install the Sun ONE Web Servers on the computers you want to include in the cluster.

   Make sure the Administration Server for the cluster has a username and password that the master Administration Server can use for authentication. You can do this either by using the default username and password or by setting up distributed administration.

2. Install the server that will contain the master Administration Server, making sure the username and password matches the one set in Step 1.
3. Add a server to the cluster list.

4. Administer a remote server by accessing its Server Manager forms from the cluster form or by copying a configuration file from one server in the cluster to another.

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

Adding a Server to a Cluster

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

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

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

1. Make sure the master Administration Server is tuned on.
2. Access the Administration Server and choose the Cluster Mgmt tab.
3. Click the Add Server link.
4. Choose the protocol that the remote Administration Server uses.
   - `http` for a normal Administration Server
   - `https` for a secure Administration Server
5. Enter the fully qualified domain name as it appears in the `magnus.conf` file of the remote server in the Admin Server Hostname field.

   For example: `plaza.sun.com`
6. Enter the port number for the remote Administration Server.

7. Click OK.

   Your master Administration Server now attempts to contact the remote server. This can take a few minutes. You will receive a message confirming the server is added to the cluster.

8. Click OK.

NOTE If you have two or more servers on different computers that use the same identifier, the server identifier and the hostname for each computer are displayed. When both server identifier and hostnames are the same, the port number is also displayed.

NOTE When you enable cluster control, the master of the cluster creates a number of files in the `https-`*`server-instance`*`/config/cluster/`*`server-name`*`/https-`*`server-name`*`/` directory for each slave in the cluster. These files are not configurable.

Modifying Server Information

Use the Modify Server option 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 also need to modify the information about that Administration Server stored in the cluster. Any other changes to the slave administration server require you to delete the server, and then add it back into the cluster after the changes have been made.

The remote administration servers will not be affected by modification to the master cluster database, unless their files have been transferred through Cluster Control.

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

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

   All servers appear listed by their unique server identifier.

3. Select the server or server to modify by:
   - Checking a specific server

- Clicking Select All

Click Reset to undo all selections.

4. Enter the new port number.
5. Click OK.

Removing Servers from a Cluster

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

1. Go to the master Administration Server and choose the Cluster Mgmt tab.
2. Click the Remove Server link.
3. Select the remote server or servers to modify by:
   - Checking a specific server
   - Clicking Select All

   Click Reset Selection to undo all selections.
4. Click OK.

A message appears confirming that the server is removed from the cluster. You can no longer access the removed server through the cluster; you can only access it now through it's own Administration Server.

Controlling Server Clusters

Sun ONE Web Server 6.1 allows you to control the remote servers in your cluster by:

- Starting and stopping them
- Viewing their access and error logs
- Transferring configuration files to them.

CAUTION Clusters must be homogeneous. All servers in the cluster must be either UNIX or Windows. Transferring configuration files from a different platform may cause the server to hang or crash.

To control servers within your cluster, perform the following steps:

1. Go to the Server Manager for the master Administration Server, and choose the Cluster Mgmt tab.
2. Click the Cluster Control link.
3. Select the server or servers to control by:
   - Checking a specific server
   - Clicking Select All to select all of the servers in the cluster

   Click Reset Selection to undo all selections.
4. Select Start or Stop remote servers from the drop down menu.
5. Select View Access or View Error log records from the drop down menu and enter the number of lines you wish to view.
6. To transfer configuration files:
   a. Select the configuration file you want to transfer in the drop down menu
   b. Select server you want to transfer it from in the drop down menu
   c. Click Transfer.

Adding Variables

Variables are used when servers in a cluster need to be configured with different values. These values might be macros to define slaves using different port numbers, or plug-ins to define different `shlib` paths.

Adding variables affects only the master cluster database. The remote administration servers will not be affected unless their files have been transferred through Cluster Control. When variables are defined, the Administration Server can no longer run independently.

To add variables for a remote server within your cluster, perform the following steps:

1. From the master Administration Server, and choose the Cluster Mgmt tab.
2. Click the Add Variables link.
3. Check the specific server you wish to add variables for.

4. In the Name field enter the type of variable you are adding.

   For example: ‘Port’.

5. In the Value field enter the value you are adding.

   For example: if ‘Port’ is entered in the name field, the value would be the port number.

6. Click OK.

   A message appears confirming that the server variable has been added.

7. Click OK.

The variable must also be added to the server's configuration file you are transferring to the slave. For example, if you are transferring the variable `port`, the variable should be declared in a server configuration file, say `server.xml`, as shown below:

```
<SERVER legacyls="ls1" qosactive="no" qosmetricsinterval="30"
qosrecomputeinterval="100">

...

<LS id="ls1" ip="0.0.0.0" port="$port" security="off"
acceptorthreads="1" blocking="no">

...

</SERVER>
```

You can set variables with different values for each slave in the configuration file. Once added, variables can also be edited and deleted using the drop-down Option list in the Add Variables page.

Part 3

Configuring, Monitoring, and Performance Tuning

Chapter 8

Configuring Server Preferences

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

This chapter contains the following sections:

Starting and Stopping the Server

Once the server is installed, it runs constantly, listening for and accepting HTTP requests.

The status of the server appears in the Server On / Off page. You can start and stop the server using one of the following methods:

- Click the Server On or Server Off in the Server On / Off page.
- Use the Services window in the Control Panel (Windows).

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

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

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

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.

NOTE On UNIX, some Sun ONE Web Server installations may require access to more memory and/or file descriptors than your operating system allows by default. If you are unable to start the server, check the resource limits imposed by your operating system using the `ulimit` command. Your operating system's `ulimit` man page should provide more information.

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, which can be found in *`server_root`*`/https-`*`server_name`*`/config/`. By default it is set to 30 seconds. To change the value, add the following line to `magnus.conf`:

```
TerminateTimeout seconds
```

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

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

Restarting the Server (UNIX/Linux)

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

- Automatically restart it from the `inittab` file.

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

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

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

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

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

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

Starting SSL-enabled Servers Automatically

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

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

2. 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 `netscape`, the edited line might look like this:

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

Restarting With Inittab (UNIX/Linux)

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

```
http:23:respawn:server_root/type-identifier/start -start -i
```

where *`server_root`* is the directory where you installed the server, and *`type-identifier`* 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.

Restarting With the System RC Scripts (UNIX/Linux)

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

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

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

Restarting the Server Manually (UNIX/Linux)

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

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

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.

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

Stopping the Server Manually (UNIX/Linux)

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

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

server_root`/type-identifier/stop`

Restarting the Server (Windows)

You can restart the server by:

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

For Windows, perform the following steps:

1. In the Control Panel double-click the Services icon.
2. Scroll through the list of services and select the service for your server.
3. Check Automatic to have your computer start the server each time the computer starts or reboots.
4. Click OK.

NOTE You can also use the Services dialog box to change the account the server uses. For more information about changing the account the server uses, see "Changing the User Account (UNIX/Linux)" on page 100.

By default, the web server prompts the administrator for the key database password before starting up. If you want to be able to restart an unattended web server, you need to save the password in a `password.conf` file. Only do this if your system is adequately protected so that this file and the key databases are not compromised.

Using the Automatic Restart Utility (Windows)

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

Changing the Time Interval (Windows)

To change the time interval that elapses between startup and the time the server can automatically restart, perform the following steps:

1. Start the Registry Editor.
2. Select your server's key (in the left side of the Registry Editor window, located in `HKEY_LOCAL_MACHINE\SOFTWARE\Netscape\Enterprise\6.0`).
3. Choose Add Value from the Edit menu. The Add Key dialog box appears.
4. In Value Name, type `MortalityTimeSecs`.
5. Select REG_DWORD from the Data Type drop-down list.
6. Click OK. The DWORD Editor dialog box appears.
7. Type the time interval (in seconds) that will elapse between startup and the time the server can restart automatically.

   The interval can be in binary, decimal, or hexadecimal format.
8. Click the numerical format for the value you entered in the previous step (binary, decimal, or hexadecimal).
9. Click OK.

   The `MortalityTimeSecs` value appears in hexadecimal format at the right side of the Registry Editor window.

Turning Off the Debugging Dialog Box (Windows)

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

To turn off the debugging dialog box that appears if the server crashes, perform the following steps:

1. Start the Registry Editor.
2. Select the AeDebug key, located in the left side of the Registry window in `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion`.
3. Double-click the Auto value in the right side of the window.

   The String Editor dialog box appears.
4. Change the string value to 1.

Tuning Your Server for Performance

There are two ways to tune the thread limit: through editing the `magnus.conf` file and through the Server Manager.

If you edit the `magnus.conf` file, `RqThrottleMinPerSocket` is the minimum value and `RqThrottle` is the maximum value.

The minimum limit is a goal for how many threads the server attempts to keep in the `WaitingThreads` state. This number is just a goal. The number of actual threads in this state may go slightly above or below this value. The default value is 48. The maximum threads represents a hard limit for the maximum number of active threads that can run simultaneously, which can become a bottleneck for performance. The default value is 128.

If you use the Server Manager, follow these steps:

1. Go to the Preferences tab.
2. Click the Performance Tuning link.
3. Enter the desired value in the Maximum simultaneous requests field.

For additional information, see the online help for the Performance Tuning page.

Editing the magnus.conf File

When the Sun ONE Web Server starts up, it looks in a file called `magnus.conf` in the *server_root*/*server_id*/`config` directory to establish a set of global variable settings that affect the server's behavior and configuration. Sun ONE Web Server executes all the directives defined in `magnus.conf`. You can edit certain settings in the `magnus.conf` file using the Magus Editor in the Server Manager.

For a complete description of the `magnus.conf` file and information about editing the file using a text editor, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference* and the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide*.

To access the Magnus Editor, perform the following steps:

1. Access the Server Manager and choose the Preferences tab.
2. Click the Magnus Editor link.
3. Select the settings to edit from the drop-down list and click Manage.

   The Server Manager displays the editor for the settings you specified.
4. Make the desired changes to the settings and click OK.

For more information about each Settings page, see the Magnus Editor page in the online help.

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 virtual server. When you install Sun ONE Web 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 HTTP server port number during installation (the default is 80). You cannot delete the default listen socket.

You can edit your server's listen socket settings using the Server Manager's Listen Sockets Table. To access the table, perform the following steps:

1. Access the Server Manager and click the Preferences tab.
2. Click the Edit Listen Sockets link.
3. Make the desired changes and click OK.

Choosing MIME Types

The Mime Types page allows you to edit your server's MIME files.

MIME (Multi-purpose Internet Mail Extension) types control what types of multimedia files your mail system supports. MIME types also specify what file extensions belong to certain server file types, for example to designate what files are CGI programs.

You don't need to create a separate MIME types file for each virtual server. Instead, you create as many MIME types files as you need and associate them with a virtual server. One MIME types file, `mime.types`, exists by default on the server, and cannot be deleted. This file can be the absolute path.

To access the MIME Types page, perform the following steps:

1. Access the Server Manager and click the Preferences tab.
2. Click the MIME Types link.
3. Make the desired changes and click OK.

For more information, see the Mime Settings page in the online help and Chapter 13, "Using Virtual Servers".

Restricting Access

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

You can set access control globally for all servers through the Administration Server or for a resource within a specific server instance through the Server Manager. For more information about setting access control for a resource, see .

"Setting Access Control" on page 192 in Chapter 9, "Controlling Access to Your Server".

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

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

1. Access the Server Manager and choose the Preferences tab.
2. Click the Restrict Access link.

For more information, see Chapter 9, "Controlling Access to Your Server" and the Restrict Access page in the online help.

Restoring Configuration Settings

The Restore Configuration page allows you to view a backup copy of your configuration files and revert to the configuration data saved on a specific date.

NOTE On Windows, use this page only to roll back your own changes to the configuration files. Do not roll back to backup versions created during installation; they may not be complete.

For more information, see the Restore Configuration page in the online help.

Configuring the File Cache

The Sun ONE Web Server uses a file cache to serve static information faster. In the previous version of the server, there was also an accelerator cache which routed requests to the file cache, but the accelerator cache is no longer used. The file cache contains information about files, and static file content. The file cache also caches information that is used to speed up processing of server-parsed HTML.

The file cache is turned on by default. The file cache settings are contained in a file called `nsfc.conf`. You can use the Server Manager to change the file cache settings.

For more information, see the online *Performance Tuning and Sizing Guide* on `http://docs.sun.com`.

Adding and Using Thread Pools

You can use thread pools to allocate a certain number of threads to a specific service.

Another use for thread pools is for running thread-unsafe plugins. By defining a pool with the maximum number of threads set to 1, only one request is allowed into the specified service function.

When you add a thread pool, the information you specify includes the minimum and maximum number of threads, the stack size, and the queue size.

For more information, see the online *Performance Tuning and Sizing Guide* on `http://docs.sun.com`.

The Native Thread Pool and Generic Thread Pools (Windows)

On Windows, you can use two types of thread pools: the native thread pool (`NativePool`) and additional generic thread pools.

To edit the native thread pool, access the Native Thread Pool page in the Server Manager.

You can create as many generic thread pools as you want, for as many purposes as you want. To create generic thread pools, access the Generic Thread Pools page in the Server Manager.

Thread Pools (UNIX/Linux)

Since threads on UNIX/Linux are always OS-scheduled (as opposed to user-scheduled) UNIX/Linux users do not need to use the `NativePool`, and do not have a Server Manager page for editing its settings. However, UNIX/Linux users can still create thread pools. To create thread pools, access the Thread Pools page in the Server Manager.

Editing Thread Pools

Once you have added a thread pool, you can change the values of the thread pool settings (minimum threads, maximum threads and so on) through the Server Manager.

You can also edit the thread pool settings in *vsclass*.obj.conf, where *vsclass* is the virtual server class name.

A thread pool appears in *vsclass*.obj.conf as follows:

```
Init fn="thread-pool-init" name=name_of_the_pool MaxThreads=n
MinThreads=n QueueSize=n StackSize=n
```

Use the following parameters to change the pool: MinThreads, MaxThreads, QueueSize, and StackSize.

Windows users can always edit the settings for the native pool using the Server Manager.

Using Thread Pools

After you've set up a thread pool, use it by designating it as the thread pool for a specific service.

To configure a thread pool, go to the Server Manager Preferences tab and select Thread Pool.Once a thread pool is configured, then the Thread Pool list will show the thread pool available to be used for the specific service you've designated

You can also designate a thread pool by using the pool parameter of the load-modules function in *vsclass*.obj.conf, where *vsclass* is the virtual server class name.

```
pool="name_of_pool"
```

In addition, you can use the pool parameter on any NSAPI function so that only that NSAPI function runs on the pool you specify.

Chapter 9

Controlling Access to Your Server

This chapter discusses the various methods you can use to control access to the Administration Server and to the files or directories on your web site. For example, for the Administration Server, you can specify who has full control of all the servers installed on a machine and who has partial control of one or more servers. Before you can use access control on the Administration Server, you must enable distributed administration from and set up an administration group in your LDAP database. This chapter assumes you have already configured distributed administration and have defined users and groups in your LDAP database.

You should also ensure the security of the web server as discussed in Chapter 4, "J2EE-based Security for Web Container and Web Applications", and in Chapter 6, "Using Certificates and Keys".

This chapter contains the following sections:

What Is Access Control?

Access control allows you to determine:

- Who can access Sun ONE Web Administration Server
- Which programs they can access
- Who can access the files or directories on your web site

You can control access to the entire server or to parts of the server, or the files or directories on your web site. You create a hierarchy of rules called access control entries (ACEs) to allow or deny access. Each ACE specifies whether or not the server should check the next ACE in the hierarchy. The collection of ACEs you create is called an access control list (ACL).

By default, the server has one ACL file that contains multiple ACLs. After determining the virtual server to use for an incoming request, Sun ONE Web Server checks if any ACLs are configured for that virtual server. If ACLs are found that apply for the current request, Sun ONE Web Server evaluates their ACEs to determine whether access should be granted or denied.

You allow or deny access based on:

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

Setting Access Control for User-Group

You can limit access to your web server to certain users or groups. User-Group access control requires users to enter a username 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 the basic authentication. If you wish to require client authentication on your 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 details, see the section Configuring a Directory Service. The information that 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, it supports the following types of User-Group authentication methods for server instances:

- Default
- Basic
- SSL
- Digest
- Other

When the server uses an internal file-based directory service, the User-Group authentication methods for server instances it supports include:

- Default
- Basic
- Digest

User-Group authentication requires users to authenticate themselves before getting access to the Administration Server, or the files and directories on your web site. With authentication, users verify their identity by entering a username and password, using a client certificate, or digest authentication plug-in. Using client certificates requires encryption. For information on encryption and using client certificates, see Chapter 4, “J2EE-based Security for Web Container and Web Applications”.

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 you check Default, the ACL rule doesn’t specify a method in the ACL file. Choosing Default allows 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 username and password to access your web server or web site. It is the default setting. You must create and store a list of users and groups in an LDAP database, such as the Sun ONE Directory Server, or in a file. You must use a directory server installed on a different server root than your web server, or a directory server installed on a remote machine.

When users attempt to access a resource that has User-Group authentication in the Administration Server or on your web site, the web browser displays a dialog box asking the user to enter a username and password. The server receives this information encrypted or unencrypted, depending on whether encryption is turned on for your server.

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

The following dialog appears when users authenticate themselves to the server:

Example of Username and Password Prompt

After clicking OK, the user will see:

- The Server Administration page, if authenticated to access Sun ONE Web Administration Server
- The file or directory listing requested, if logging in to a web site
- A message denying access if the username or password was invalid

You can customize the access denied message that unauthorized users receive in the Access Denied Response page.

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 you set the server to use certificate information for authenticating the client, the server:

- Checks first if the certificate is from a trusted CA. If not, the authentication fails and the transaction is ended. To learn how to turn on client authentication, see "Requiring Client Authentication" on page 138.
- Maps the certificate to a user's entry using the `certmap.conf` file, if the certificate is from a trusted certificate authority (CA). To learn how to set up the certificate mapping file see "Using the certmap.conf File" on page 141.
- Checks the ACL rules specified for that user if the certificate maps correctly. Even if the certificate maps correctly, ACL rules can deny the user access.

Requiring client authentication for controlling access to specific resources differs from requiring client authentication for all connections to the server. If you set the server to require client authentication for all connections, the client only needs to present a valid certificate issued by a trusted CA. If you set the server's access control to use the SSL method for authentication of users and groups, the client will need to:

- 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, you need to have SSL ciphers enabled for your web server. See Chapter 6, "Using Certificates and Keys" to learn how to enable SSL.

In order to successfully gain access to an SSL authenticated resource, the client certificate must be from a CA trusted by the web server. The client certificate needs to be published in a directory server if the web 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 the `certmap.conf` file to only compare the user ID and email address in the browser certificate with the directory server entry. To learn more about `certmap.conf` and certificate mapping, see Chapter 6, "Using Certificates and Keys".

NOTE Only the SSL authentication method requires modification to the `certmap.conf` file, because the certificate is checked against the LDAP directory. Requiring client authentication for all connections to the server does not. If you choose to use client certificates, you should increase the value of the `AcceptTimeout` directive in `magnus.conf`.

Digest Authentication

Sun ONE Web Server 6.1 can be configured to perform digest authentication using either an LDAP-based or a file-based directory service.

Digest authentication allows the user to authenticate based on username and password without sending the username 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 Web 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. In order for this to work, your directory server needs access to the user's password in cleartext. Sun ONE 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 need to enable the reversible password plug-in and the digestauth-specific plug-in included with Sun ONE Web Server 6.1. To configure your web server to process digest authentication, set the `digestauth` property of the database definition in `dbswitch.conf`.

The server tries to authenticate against the LDAP database based upon the ACL method specified, as shown in Table 9-1. If you do not specify an ACL method, the server will use either digest or basic when authentication is required, or basic if authentication is not required. This is the preferred method.

Table 9-1 Digest Authentication Challenge Generation

ACL Method	Digest Authentication Supported by Authentication Database	Digest Authentication Not Supported by Authentication Database
“default”	digest and basic	basic
none specified		
“basic”	basic	basic
“digest”	digest	ERROR

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

- Checking for Authorization request header. If not found, a 401 response is generated with a Digest challenge, and the process stops.
- Checking for Authorization type. If Authentication type is Digest the server then:
  - Checks nonce. If not a valid, fresh nonce generated by this server, generates 401 response, and the process stops. If stale, generates 401 response with `stale=true`, and the process stops.

    You can configure the time the nonce remains fresh by changing the value of the parameter `DigestStaleTimeout` in the `magnus.conf` file, located in *`server_root`*`/https-`*`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 will remain fresh. After the specified seconds elapse, the nonce expires and new authentication is required from the user.
  - Checks realm. If it does not match, generates 401 response, and process stops.
  - Checks existence of user in LDAP directory if the authentication directory is LDAP-based, or checks existence of user in file database if the authentication directory is file-based. If not found, generates 401 response, and the process stops.
  - Gets request-digest value from directory server or file database and checks for a match to client's request-digest. If not, generates 401 response, and process stops.
  - Constructs 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 need to 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’re using a file-based authentication database, you don’t 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, perform the following steps:

1. Make sure this shared library resides on the same server machine that the Sun ONE Directory Server is installed on.
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 will need to copy several `.dll` files from the Sun ONE Web Server installation to your Sun ONE Directory Server server machine in order for Sun ONE Directory Server to start properly with the Digest plug-in.

To install the Digest Authentication plug-in on Windows, perform the following steps:

1. Access the shared libraries in the Sun ONE Web Server installation in:

   [*server_root*]`\bin\https\bin`
2. Copy the files:
   - `nsldap32v50.dll`
   - `libspnr4.dll`
   - `libplds4.dll`
3. Paste them into either:
   - `\Winnt\system32`
   - Sun ONE Directory Server install directory: [*server_root*]`\bin\sldap\server`

Setting the Sun ONE 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 Sun ONE Directory Server to use the DES algorithm, perform the following steps:

1. Launch the Sun ONE Directory Server Console.
2. Open your iDS 5.0 instance.
3. Select the Configuration tab.
4. Click on 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 ONE Directory Server instance.

NOTE In order 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.

Setting Access Control for Host-IP

You can limit access to the Administration Server, or the files and directories on your web site by making them available only to clients using specific computers. You specify hostnames or IP addresses for the computers that you want to allow or deny. You can use wildcard patterns to specify multiple computers or entire networks. 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 username or password.

Since 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 username and password will be required for access.

Host-IP authentication does not require DNS 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. You can enable DNS on your server through the Performance Tuning page in the Preferences tab on your Server Manager.

Enabling DNS degrades the performance of Sun ONE Web Server since the server is forced to do DNS look-ups. To reduce the effects of DNS look-ups on your server's performance, resolve IP addresses only for access control and CGI instead of resolving the IP address for every request. To do this, `iponly=1` to `AddLog fn="flex-log" name="access"` in your `obj.conf` file:

```
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 your web site, the settings are stored in a file with the extension `.acl`. Access control files are stored in the directory *install_dir*`/httpacl` with *install_dir* 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 name is `generated-https-`*server-id*`.acl`; the temporary working file is called `genwork-https-`*server-id*`.acl`. If you use Sun ONE Administration Server to configure access, you'll have these two files. However, if you want more complex restrictions, you can create multiple files, and reference them from the `server.xml` file. There are also a few features available only by editing the files such as restricting access to the server based on the time of day or day of the week.

You can also manually create and edit `.acl` files to customize access control using APIs. For more information on using access control APIs, see the *Programmer's Guide*.

For more information on access control files and their syntax, see Appendix C, "ACL File Syntax".

Configuring the ACL User Cache

By default, the Sun ONE Web Server caches user and group authentication results in the ACL user cache. You can control the amount of time that 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 Sun ONE Web Server every time you make changes to the LDAP entries. For example, if this value is set to 120 seconds, Sun ONE Web 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 will result in several LDAP directory accesses on every request.

For more information on ACL file directives, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide.*

How Access Control Works

When the server gets a request for a page, the server uses the rules in the ACL file to determine if it should grant access or not. The rules can reference the hostname or IP address of the computer sending the request. The rules can also reference users and groups stored in the LDAP directory.

For example, the following ACL file contains the two default entries for the Administration Server (`admin-serv`), plus an additional entry that allows users in the "admin-reduced" group to access the Preferences tab in the Administration Server.

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to Sun ONE Web Server.
# These "es-internal" rules should not be modified.
  acl "es-internal";
  allow (read, list, execute,info) user = "anyone";
  deny (write, delete) user = "anyone";
# The following "default" rules apply to the entire document
# directory of Sun ONE Web Server. In this example, the rules
# are set up so that "all" users in the directory server are
# allowed to read, execute, list, and get information.
# The "all" users are not allowed to write to or delete any files.
# All clients that accesses the document directory of the web
# server will be required to submit a username and password
# since this example is using the "basic" method of
# authentication. A client must be in the directory server
# to gain access to this default directory since "anyone"
# not in the directory server is denied, and "all" in the
# directory server are allowed.
  acl "default";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (user = "all");
# The following rules deny access to the directory "web"
# to everyone not in the directory server and deny everyone
# in the directory server who is not in GroupB.
# Only the users in GroupB are allowed read, execute, list,
# and info permissions. GroupA can not gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory “my_stuff”. Furthermore, members
# of GroupB can not write or delete files.
  acl "path=/export/user/990628.1/docs/my_stuff/web/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");

  allow (read,execute,list,info)
  (group = "GroupB");

# The following rule denies everyone not in the directory
# server and denies everyone in the directory server except
# user with the ID of "SpecificMemberOfGroupB". The ACL rule
# in this setting also has a requirement that the user
# connect from a specific IP address. The IP address setting
# in the rule is optional; it has been added to for extra
# security. Also, this ACl rule has a Customized prompt
# of "Presentation Owner". This Customized prompt appears
```

```
# in the username and password dialog box in the client's
# browser.

  acl
"path=/export/user/990628.1/docs/my_stuff/web/presentation.html"
;
  authenticate (user,group) {
     database = "default";
     method = "basic";
     prompt = "Presentation Owner";
  };
  deny (all)
  (user = "anyone" or group = "my_group");
  allow (all)
  (user = "SpecificMemberOfGroupB") and
  (ip = "208.12.54.76");

# The following ACL rule denies everyone not in the directory
# server and everyone in the directory server except for
# GroupA and GroupB access to the directory “my_stuff”
  acl "path=/export/user/990628.1/docs/my_stuff/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (group = "GroupA,GroupB");
```

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

Sun ONE Web Server would first check access control for the entire server. If the ACL for the entire server was set to continue, the server would check 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 there are more than one ACLs that match, the server uses the last ACL statement that has a match. The `default` ACL is bypassed since the `uri` ACL is the last statement that matches.

Setting Access Control

This section describes the process of restricting access to the files or directories on your web site. 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 to updating data to only human resource personnel responsible for payroll.

You can set access control globally for all servers through the Administration Server. Each option is described in detail in the following section, Selecting Access Control Options.

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

Setting Access Control Globally

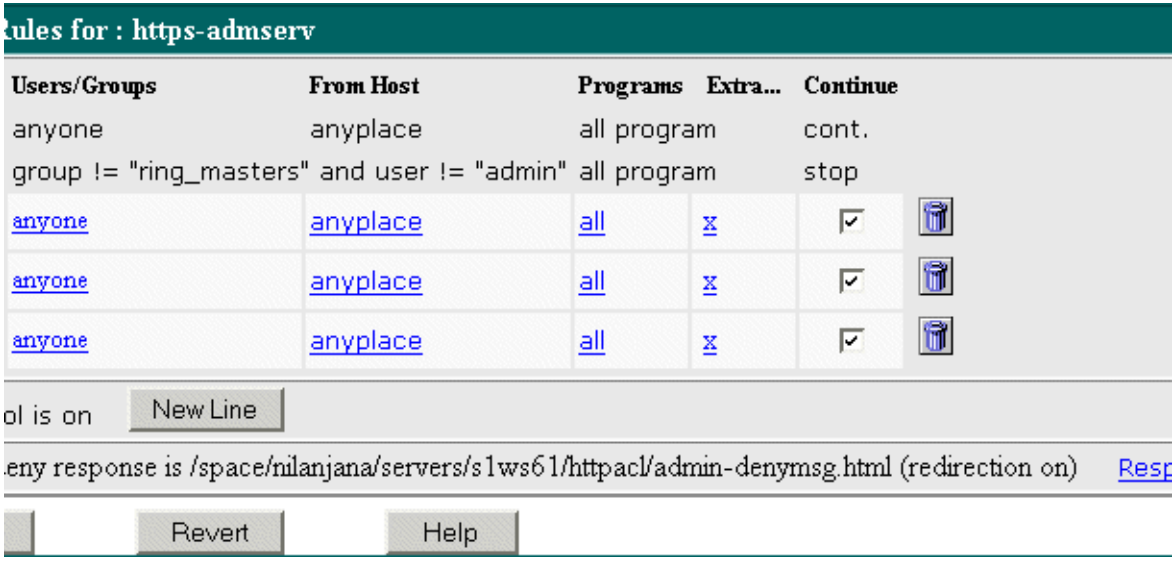
To create or edit access control globally for all servers, perform the following steps:

1. Access the Administration Server and choose the Global Settings tab.
2. Click the Restrict Access link.
3. Select the administration server (https-admserv) from the drop-down list.

4. Click Create ACL and the Go button.

   The Access Control Rules for uri=/https-admserv/ page appears:

   Access Control Rules Page.



   The Administration Server has two lines of default access control rules which cannot be edited.

5. Check 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.

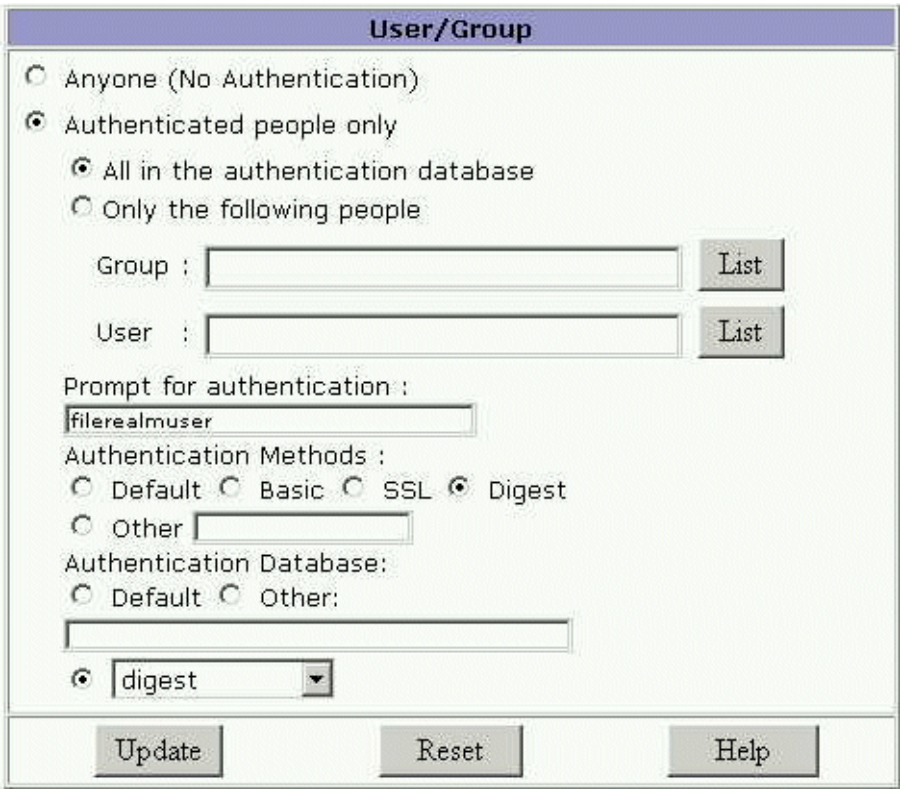
   To swap an access control restriction with the access control restriction preceding it, click the up arrow figure.

   To swap an access control restriction with the access control restriction after it, click the down arrow figure.

7. Click on anyone in the Users / Groups column.

   The User / Group page appears in the lower frame:

   User / Group Page



8. Select which users and groups you will allow access to and click Update.

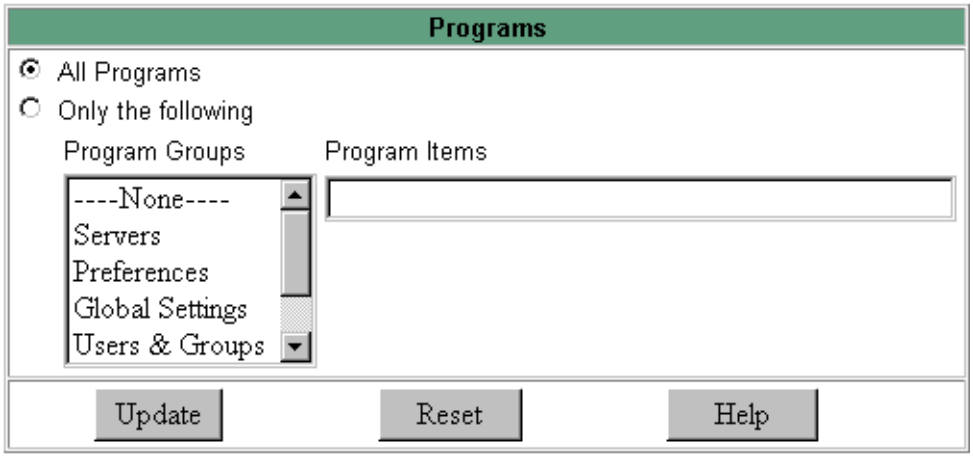
   Clicking List for Group and User will provide lists for you to choose from.

9. Click on anyplace in the From Host column.

10. Enter Host Names and IP Addresses allowed access and click Update.

11. Click on all programs in the Programs column.

    Programs



12. Select the Program Groups or enter the specific file name in the Program Items field you will allow access to, and click Update.
13. (Optional) Click the x under the Extra column to add a customized ACL expression.
14. Put a check in the Continue column, if it isn't already selected as the default.

    The server will evaluate 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 ones.
15. (Optional) Click Response when denied to direct the user to a different URL or URI.
16. Enter the path to the absolute URL or a relative URI and click update.
17. Click Submit to store the new access control rules in the ACL file.

NOTE Clicking Revert will remove all of the settings you've just created.

Setting Access Control for a Server Instance

You can create, edit, or delete access control for a specific server instance using the Server Manager.

NOTE If deleting, you should not delete all the 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.

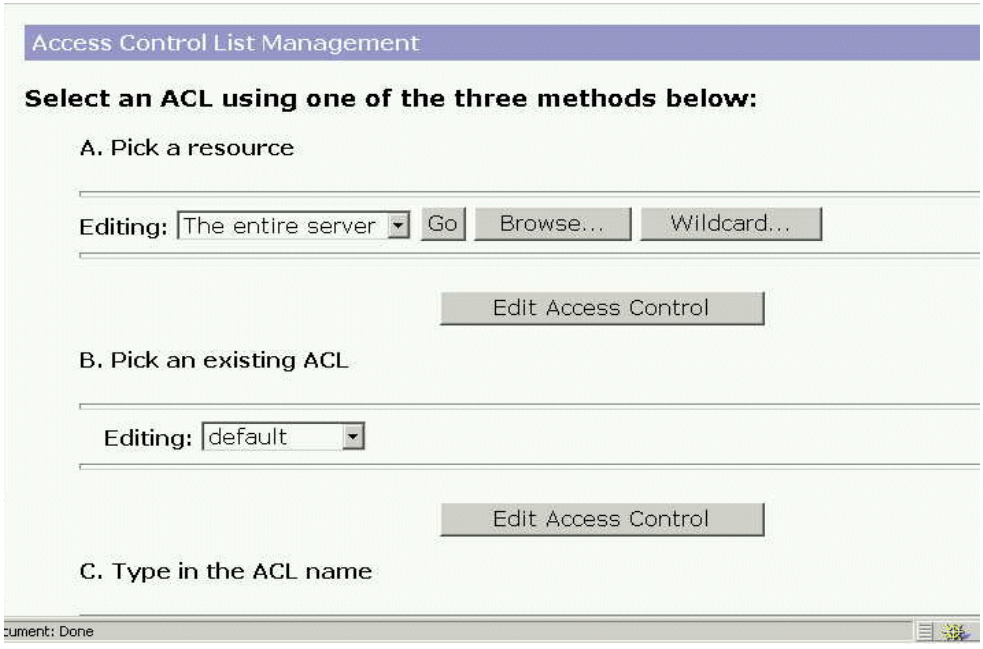
To create access control for a server instance, perform the following steps:

1. Access the Server Manager and select the server instance you wish to create or edit ACLs for.
2. Choose the Preferences tab from the Server Manager.
3. Click the Restrict Access link.
4. Under the Option column choose one of the following:
   - Add and enter the ACL file location
   - Edit and select the ACL file from the drop-down menu

- Delete from the drop-down menu and select the ACL file

The Access Control List Management Page offering three options appears:

Access Control List Management Page



5. Select one of the following:

   - Pick a resource to specify a wildcard pattern for files or directories (such as `*.html`), choose a directory or a filename to restrict, or browse for a file or directory.
   - Pick an existing ACL to select from a list of all the ACLs you have enabled. Existing ACLs you have not enabled will not appear in this list.

- **Enter the ACL name** allows to create named ACLs. Use this option only if you're familiar with ACL files. You'll need to manually edit `obj.conf` if you want to apply named ACLs to resources.

Table 8-2 describes the resource wildcards you can use.

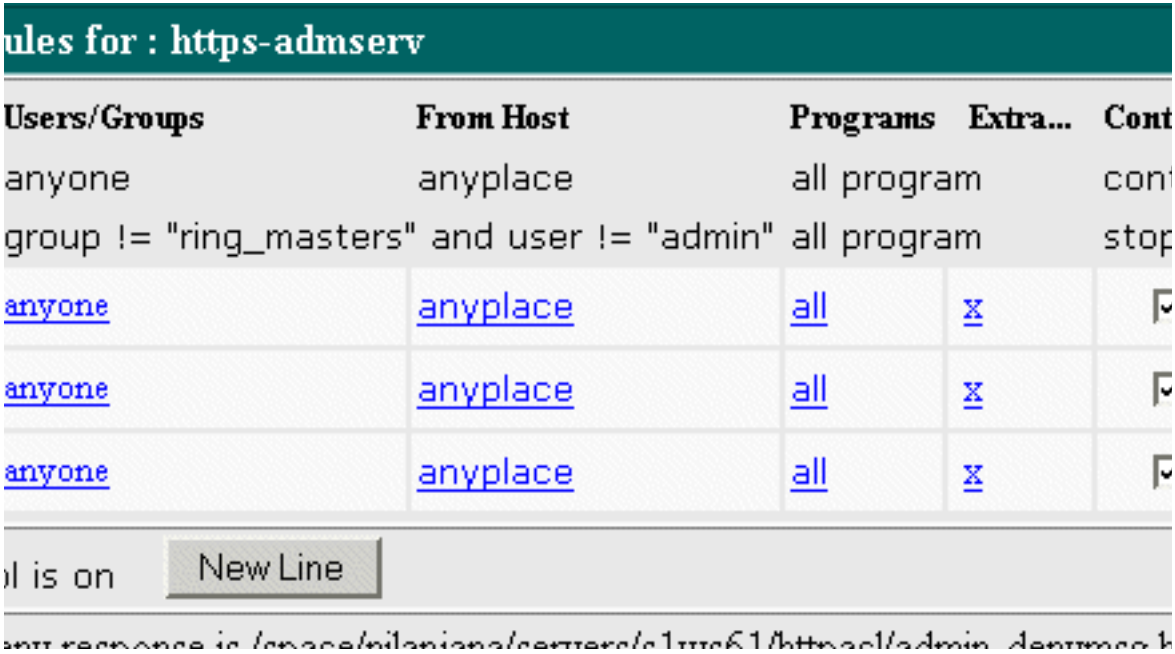
Table 9-2 Server Resource Wildcards

Resource wildcard	What it means
default	A named ACL created during installation that restricts write access so only users in the LDAP directory can publish documents.
Entire Server	One set of rules determines the access to your entire web site, including any virtual servers you have running. To restrict access to a virtual server, specify the path of its document root.
`/usr/sun/server4/docs/cgi-bin/*`	Controls access to all files and directories in the `cgi-bin` directory. You must specify an absolute path. On Windows, the path must include the drive letter.
`uri="/sales"`	Controls access to the `sales` directory in the document root. To specify URIs, create a named ACL.

6. Click Edit Access Control.

   The Access Control Rules for: (server instance) appears.

   Access Control Rules Page

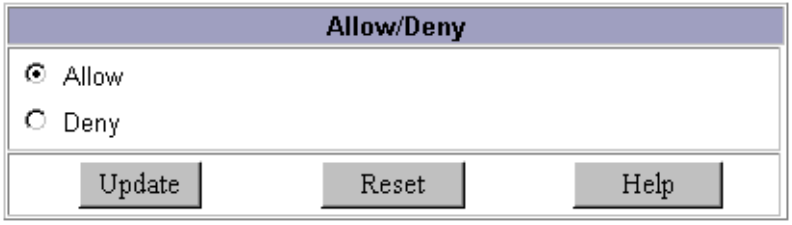


7. Check Access control is on, if not already selected.

8. To create or edit the ACL for this server instance, click on Deny in the Action column.

   The Allow / Deny page is displayed in the lower frame:

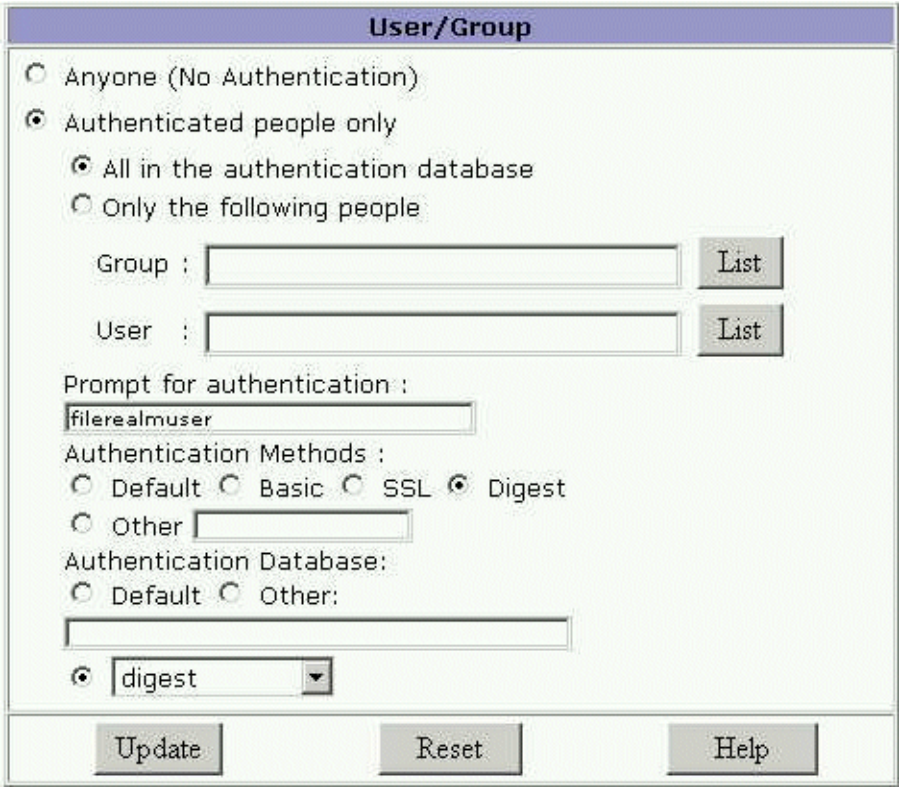
   Allow / Deny Page



9. Select Allow, if it isn't already selected as the default, and click Update.
10. Click on anyone in the Users / Groups column.

    The User / Group page appears in the lower frame:

    User / Group Page

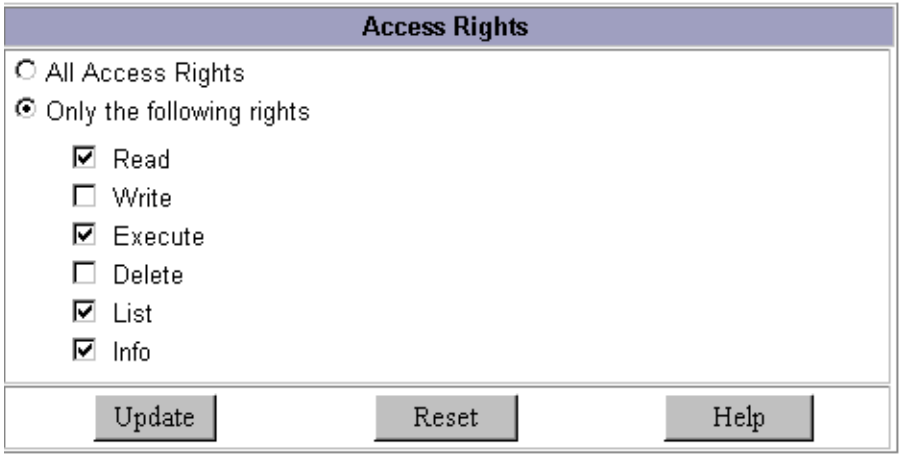


11. Select which users and groups you will allow access to and click Update.

    Clicking List for Group and User will provide lists for you to choose from.
12. Click on anyplace in the From Host column.
13. Enter Host Names and IP Addresses allowed access and click Update.

14. Click on all in the Rights column.

    Access Rights Page



15. Select one of the following and then click Update:
    - All Access Rights
    - Only the following rights and check all appropriate rights for this user
16. (Optional) Click the x under the Extra column to add a customized ACL expression.
17. Put a check in the Continue column, if it isn't already selected as the default.

    The server will evaluate 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 ones.
18. (Optional) Click Response when denied to direct the user to a different URL or URI.
19. Enter the path to the absolute URL or a relative URI and click update.
20. Click Submit to store the new access control rules in the ACL file.

NOTE Clicking Revert will remove all of the settings you've just created.

21. Repeat all steps above for each server instance you wish to establish access control for.
22. When finished, click Apply.
23. Select hard start /restart or dynamically apply.

ACL settings can also be enabled on a per virtual server basis. To learn how this is done, see "Editing Access Control Lists for Virtual Servers" on page 226.

Selecting Access Control Options

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

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 expressions (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* checked, everyone would be denied access to the resource. The server continues down the list until it reaches either an ACE that doesn't match, or that matches but is set to not continue. The last matching ACE determines if access is allowed or denied.

Specifying Users and Groups

With user and group authentication, users are prompted to enter a username and password before they can access the resource specified in the access control rule.

Sun ONE Web Server checks lists of users and groups stored either in an LDAP server, such as Sun ONE 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.

- **Anyone (No Authentication)** is the default and means anyone can access the resource without having to enter a username or password. However, the user might be denied access based on other settings, such as host name or IP address. For the Administration Server, this means that anyone in the administrators group that you specified with distributed administration can access the pages.
- **Authenticated people only**
  - **All in the authentication database** matches any user who has an entry in the database.
  - **Only the following people** lets you specify 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** allows you to enter message text that appears in the authentication dialog box. You can use this text to describe what the user needs to enter. Depending on the operating system, the user will see about the first 40 characters of the prompt. Netscape Navigator and Netscape Communicator cache the username and password, and associate them with the prompt text. When the user accesses files and directories of the server having the same prompt, the usernames and passwords won't need to be entered again. If you want users to authenticate again for specific files and directories, you simply need to 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.
  - **Default** uses the default method you specify in the `obj.conf` file, or "Basic" if there is no setting in `obj.conf`. If you check Default, the ACL rule doesn't specify a method in the ACL file. Choosing Default allows 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 username and password are only encrypted if encryption is turned on for the server.

- **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, you can combine Basic and SSL methods.
- **Digest** uses the an authentication mechanism that provides a way for a browser to authenticate based on username and password without sending the username 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 Web 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** lets you select a 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 wish to configure individual ACLs to use different databases, select Other, and choose the database from the drop-down list. Non-default databases and LDAP directories need to have been specified in the file *server_root*`/userdb/dbswitch.conf`. If you use the access control API for a custom database, such as Oracle or Informix, select Other, and enter the database name.

Specifying the From Host

You can restrict access to the Administration Server or your web site based on which computer the request comes from.

- **Anyplace** allows access to all users and systems
- **Only from** allows you to restrict access to specific Host Names or IP Addresses

If you select the Only from option, enter a wildcard pattern or a comma-separated list in the Host Names or IP Addresses fields. Restricting by hostname is more flexible than by IP address: if a user's IP address changes, you won't need to update this list. Restricting by IP address, however, is more reliable: if a DNS lookup fails for a connected client, hostname restriction cannot be used.

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 `*.sun.com`. You can set different hostnames and IP addresses for superusers accessing the Administration Server.

For hostnames, the * must replace an entire component of the name. That is, `*.sun.com` is acceptable, but `*users.sun.com` is not. When the * appears in a hostname, it must be the left-most character. For example, `*.sun.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 * appears in an IP address, it must be the right-most character. For example, `198.*` is acceptable, but not `198.*.251.30`.

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 they can configure that server. For example, you might allow some administrators to configure the Users & Groups section of the administration server and not allow them access to the Global Settings.

You can configure different users to access different functional domains. Once a user is setup with access to a few selected functional domains, after the user logs in, Administration Server pages from only those functional domains for which you have granted access to that user are visible.

- **All Programs** allows or denies access to all programs. By default administrators have access to all programs for a server.
- **Only the following Program Groups** allows you to specify which programs the user has access to. Select the program from the drop-down list. You can choose multiple program groups by pressing the Control key while clicking on the groups. You can restrict access to the following programs groups:
  - None (default)
  - Servers
  - Preferences
  - Global Settings
  - Users & Groups
  - Security
  - Cluster Mgmt

The Program Groups listed 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 username, host, and IP to determine what pages they can view.

- **Program Items** allows you to enter a page name in the Program Items field to control access to a specific page within a program.

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 web site. In addition to allowing or denying all access rights, you can specify a rule that allows or denies partial access rights. For example, you allow users read-only access rights to your files, so they can view the information, but not change the files.

- **All Access Rights** is the default and will allow or deny all rights
- Only the following rights allow you to select a combination of rights to be allowed or denied:
  - **Read** allows users to view files, including includes the HTTP methods GET, HEAD, POST, and INDEX
  - **Write** allows 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** allows users to execute server-side applications, such as CGI programs, Java applets, and agents
  - **Delete** allows users who also have write privileges to delete files or directories.

- **List** allows users to access lists of the files in directories that don't contain an `index.html` file.
- **Info** allows users to receive information about the URI, for example `http_head`.

Writing Customized Expressions

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

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

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

For more information on valid syntax and ACL files, see Appendix C, "ACL File Syntax" and "Referencing ACL Files in obj.conf" on page 472.

Turning Off Access Control

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

If you want to deactivate an ACL, you can comment out the ACL lines in the file `generated-https-server-id.acl` by putting # signs at the beginning of each line.

From the Administration Server, you could create and turn on access control for a specific server instance and leave it off (which is the default) for other servers. For example, you could deny all access to the Server Manager pages from the Administration Server. With distributed administration on and access control off by default for any other servers, administrators could still access and configure the other servers, but they cannot configure the Administration Server.

NOTE This access control is in addition to the user being in the administrators group set for distributed administration. The Administration Server first checks that a user (other than superuser) is in the administrators group, and then evaluates the access control rules.

Responding When Access is Denied

Sun ONE Web Server provides the following default message when access is denied: “FORBIDDEN. Your client is not allowed access to the restricted object.” You can choose a different response when denied access. You can also create a different message for each access control object.

To change the message sent for a particular ACL, perform the following steps:

1. Click the Response when denied link in the ACL page.
2. Check Respond with the following file in the lower frame.
3. Enter the path to the absolute URL or a relative URI and click update.

   Make sure users have access to the URL or URI they are redirected to.
4. Click Update.
5. Click Submit in the top frame to submit the access control rule.

Limiting Access to Areas of Your Server

This section describes some commonly used access restrictions to a web server and its contents. The steps for each procedure detail the specific actions you need to take; however, you will still need to complete all of the steps described under “Setting Access Control for a Server Instance” on page 196.

The following procedures are described in this section:

- Restricting Access to the Entire Server
- Restricting Access to a Directory (Path)
- Restricting Access to a URI (Path)
- Restricting Access to a File TypeRestricting Access Based on Time of Day
- Restricting Access Based on Security

Restricting Access to the Entire Server

You may wish to allow access to users in a group called 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.

Using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Choose the ACL File to edit.
5. Pick the entire server resource, and click Edit Access Control.
6. Add a new rule to deny access to all.
7. Add another new rule to allow access to a specific group.
8. Enter a wildcard pattern for the host names of the computers to be allowed.

   For example, `*.employee.sun.com`
9. Unselect Continue.
10. Submit and Apply your changes.

Restricting Access to a Directory (Path)

You can allow users in a group to read or run applications in directories, and its 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 limit access to a directory on the server, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Choose the ACL File to edit.

5. Browse the Pick a Resource section and select the directory you want to restrict.

   The directories in the server's document root are displayed. Once selected, the Editing drop-down list displays the absolute path to the directory.

NOTE If you want to view all files in your server root, click Options and then check List files as well as directories.

6. Click Edit Access Control.
7. Create a new rule and leave the defaults to deny access to everyone from everywhere.
8. Create another new rule allowing users in a specific group to have read and execute rights only.
9. Create a third line to allow a specific user to have all rights.
10. Unselect Continue for the second and third lines and click Update.
11. Submit and Apply your changes.

An absolute path to the file or directory would be created in the docroot directory. The entry in the ACL file would appear as follows:

```
acl "path=d:\sun\suitespot\docroot1\sales/";
```

Restricting Access to a URI (Path)

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

To limit access to a URI, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.

4. Enter the URI you want to restrict in the Type in the ACL name section.

   For example: `uri=/my_directory`.

5. Click Edit Access Control.
6. Create a new rule to allows all users read access.
7. Create another new rule to allow access for the owner of the directory.
8. Uncheck Continue for both the first and second rules.
9. Click Submit and Apply your changes.

A path for the URI is created relative to the document root. The entry in the ACL file would appear as follows: `acl "uri=/my_directory";`

Restricting Access to a File Type

You can limit access to file types on your server or web site. For example, you might wish 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 limit access to a file type, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Click Wildcard in the Pick a resource section and enter a wildcard pattern.

   For example, `*.cgi`.

5. Click Edit Access Control.
6. Create a new rule to allow read access to all users.
7. Create another rule that allows write and delete access only to a specified group.
8. Submit and Apply your changes.

For file type restriction, you would leave both continue boxes checked. If a request for a file comes in, the server will then check 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 or during specified hours or on specified days. You might use this to prevent people from publishing documents during working hours when people might be accessing the files.

To limit access based on time of day, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Select the entire server from the drop-down list in Pick a Resource and click Edit Access Control.
5. 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 won’t apply and the server will search for another rule that matches.
6. Create another new rule denying write and delete rights to all.
7. Click X link to create a customized expression.
8. Enter the days of the week and the times of day to be allowed.

   Example:

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

   The message “Unrecognized expressions” will be displayed in the Users/Groups and From Host fields when you create a custom expression.
9. Submit and Apply your changes.

Any errors in the custom expression will generate an error message. Make corrections and submit again.

Restricting Access Based on Security

As of Sun ONE Web Server 6.1 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 limit access based on security, using the steps described for setting access control for a server instance, you would:

1. Use the Server Manager to select the server instance.
2. Choose the Preferences tab.
3. Click the Restrict Access link.
4. Select the entire server from the drop-down list in Pick a Resource and click Edit Access Control.
5. 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 won't apply and the server will search for another rule that matches.
6. Create another new rule denying write and delete rights to all.
7. Click X link to create a customized expression.
8. Enter `ssl=”on”`.

   Example:

   ```
   user = "anyone" and ssl=”on”
   ```

9. Submit and Apply your changes.

Any errors in the custom expression will generate an error message. Make corrections and submit again.

Securing Access Control With Distributed Administration

This section lists the additional tasks you need to perform in order to secure access control with Sun ONE Web Server 6.1, after enabling distributed administration.

Securing Access to Resources

The order in which the PathCheck directive occurs in the `https-`*`server-id`* object tag in the `generated.https-`*`server-id`*`.acl` file might grant undesired access to resources. To prevent this, edit the `<`*`server-root`*`>/generated.https-`*`server-id`*`.acl` file, specifying a comma-separated list of program groups for which access control is required, as shown below:

Below the line:

```
allow (all)
```

`user=<`*`username`*`>` and `program=<`*`program group, program group...`*`>;`

add the following line:

```
deny absolute (all)
```

`user=<`*`username`*`>` and `program!=<`*`program group, program group...`*`>;`

Securing Access to Server Instances

In order to configure Sun ONE Web Server 6.1 to control access to server instances, edit the `<`*`server-root`*`>/httpacl/*.https-admserv.acl` files to specify the user to whom you want to grant access control privileges. Example:

```
acl "https-<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 Administration Server related ACL files (`gen*.https-admserv.acl`), then 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, then complete only step (1) below for that particular ACL.

1. Edit the `<server-root>/httpacl/gen*.https-admserv.acl` files to add `ip` to the authentication list, in addition to `user` and `group`, as shown below:

   ```
   acl "https-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";
   ```

   Example:

   ```
   acl "https-admserv";

   authenticate (user,group,ip) {

   database = "default";

   method = "basic";

   };

   deny absolute (all) ip !="205.217.243.119";
   ```

Working with Dynamic Access Control Files

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

The following topics are described in this section:

Using .htaccess Files

Sun ONE Web Server supports `.htaccess` dynamic configuration files. You can enable `.htaccess` files either through the user interface or by manually changing the configuration files. The files that support `.htaccess` are in the *`server_root`*`/plugins/htaccess` directory. These files include a plug-in that enables you to use `.htaccess` files and a script for converting `.nsconfig` files to `.htaccess` files.

You can use `.htaccess` files in combination with the server's standard access control. The standard access controls are always applied before any `.htaccess` access control, regardless of the ordering of `PathCheck` directives. Do not require user authentication with both standard and `.htaccess` access control when user-group authentication is 'Basic' You could use SSL client authentication via the standard server access control, and also require HTTP 'Basic' authentication via an `.htaccess` file.

This section includes the following topics:

Enabling .htaccess from the User Interface

To configure your Sun ONE Web Server to use `.htaccess`, perform the following steps:

1. Access the Server Manager and select the server instance you wish to enable `.htaccess` for.
2. Click on the Class Manager link at the top of the screen.
3. Select the Content Mgmt tab.
4. Click on the `.htaccess` Configuration link.

5. Select the server to edit by:
   - Choosing the entire server or a specific server from the drop-down list
   - Choosing the directory and files to edit by clicking Browse
   - Choosing a wildcard pattern to edit by clicking Wildcard
6. Select Yes to activate .htaccess.
7. Enter the file name where you want the `.htaccess` configuration to be added.
8. Click OK.
9. When finished, click Apply.
10. Select hard start / restart or dynamically apply.

Enabling .htaccess from magnus.conf

To manually enable your sever to use the `.htaccess`, you need to first modify the server's `magnus.conf` file to load, initialize, and activate the plug-in.

1. Open `magnus.conf` in the *`server_root`*`/https-`*`server_name`*`/config` file.
2. After the other `Init` directives, add the following lines:
   - For UNIX/Linux:

     ```
     Init fn="load-modules" funcs="htaccess-init,htaccess-find"
     shlib="server_root/plugins/htaccess/htaccess.so"
     NativeThread="no"
     Init fn="htaccess-init"
     ```

   - For Windows:

     ```
     Init fn="load-modules"
     funcs="htaccess-init,htaccess-find,htaccess-register"
     shlib="server_root/plugins/htaccess/htaccess.dll"
     NativeThread="no"
     Init fn="htaccess-init"
     ```

   - For HP:

     ```
     Initfn="load-modules"
     funcs="htaccess-init,htaccess-find,htaccess-register"
     shlib="<server_root>/plugins/htaccess/htaccess.sl"
     NativeThread="no"

     Init fn="htaccess-init"
     ```

3. (Optional) Make the final line read:

   ```
   Init fn=”htaccess-init”[groups-with-users=yes]
   ```

4. Click File / Save.

5. Open `obj.conf`.

6. Add the PathCheck directive as the last directive in the object.

   a. To activate `.htaccess` file processing for all directories managed by a virtual server, add the `PathCheck` directive to the default object in the `object.conf` file:

   ```
   <Object name="default">

   ...

   PathCheck fn="htaccess-find"

   </Object>
   ```

   `.htaccess` processing should be the last `PathCheck` directive in the object.

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

7. To name your `.htaccess` files something other than `.htaccess`, you must specify the filename in the `PathCheck` directive using the following format:

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

NOTE	The next time you use the Administration Server, you will be warned that manual edits have been applied. Click Apply to accept your changes.

Subsequent access to the server will be subject to `.htaccess` access control in the specified directories. For example, to restrict write access to `.htaccess` files, create a configuration style for them, and apply access control to that configuration style. For more information, see Chapter 17, “Applying Configuration Styles”.”

Converting Existing .nsconfig Files to .htaccess Files

Sun ONE Web Server 6.1 includes the `htconvert` plug-in for converting your existing `.nsconfig` files to `.htaccess` files. The `.nsconfig` files are no longer supported. If you have been using `.nsconfig` files, you should convert them to `.htaccess` files.

When activated, `htconvert` searches the given `server.xml` files for `pfx2dir` and `document-root` directives. Each `.nsconfig` file found will be translated into an `.htaccess` file. Multiple obj.conf files can be converted depending on configuration.

NOTE If there is an existing `.htaccess` file, `htconvert` will produce an `.htaccess.new` file, and give a warning. If `.htaccess` and `.htaccess.new` already exist, the new file will be named `.htaccess.new.new`. The `.new` will be repeatedly appended.

The `htconvert` plug-in currently only supports the `RestrictAccess` and `RequireAuth` directives, and the `<Files>` wrapper. If `<Files>` other than `<Files*>` are presented, the script will give a warning and behave as though all files in the directory are to be access-controlled.

To convert your files, at the command prompt, enter the path to Perl on your system, the path to the plug-in script, and the path to your `server.xml` file. For example:

server_root`\install\perl server_root/plugins/htaccess/htconvert` *server_root*`/https-server_name/config/server.xml`

All `.nsconfig` files are converted to `.htaccess` files, but not deleted.

The `groups-with-users` option facilitates handling large numbers of users in groups. If you have many users in a group, follow these steps:

1. Revise the format of the user file format to list all the groups a user belongs to:

   ```
   username:password:group1,group2,group3,...groupn
   ```

2. Revise the `AuthGroupFile` directive to point to the same file as the `AuthUserFile`.

Alternatively, you can:

1. Remove the `AuthGroupFile` directive entirely.
2. Add the following to the `Init fn=htaccess-init` line in the `magnus.conf` file:

   ```
   groups-with-users="yes"
   ```

Using htaccess-register

The `htaccess-register` is a new function allowing you to create your own authentication methods. Like Apache you can create external authentication modules and plug them into the `.htaccess` module via `htaccess-register`. Two sample modules are provided in *server_root*`/plugins/nsapi/htaccess`.

You can use external modules to create one or more new directives. For example, you might specify the user database for authentication. The directives may not appear within `<Limit>` or `<LimitExcept>` tags.

Example of an .htaccess File

The following example shows an `.htaccess` file:

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

Supported .htaccess Directives

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

allow

Syntax

Allows from host where:

- host is all, to allow access from all client hosts
- host is all or the last part of a DNS host name
- host is a full or partial IP address

Does not need to be enclosed within a <Limit> or <LimitExcept> range but usually is.

Effect

Allows access to the specified hosts. Normally appears inside a <Limit> range.

deny

Syntax

Deny from host where:

- host is all, to deny access from all client hosts
- host is all or the last part of a DNS host name
- host is a full or partial IP address

Does not need to be enclosed in a <Limit> <LimitExcept> range but usually is.

Effect

Denies access to the specified hosts. Normally appears inside a <Limit> range.

AuthGroupFile

Syntax

AuthGroupFile filename where filename is the name of file containing group definitions in the form: groupname: user user.

Must not appear within a <Limit> or <LimitExcept> range.

Effect

Specifies that the named group file is to be used for any group definitions referenced in a require group directive. Note that if the filename specified in an AuthGroupFile directive is the same as the filename in an AuthUserFile directive, the file is assumed to contain users and groups in the format:

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

AuthUserFile

Syntax

AuthUserFile filename where:

- filename is the name of file containing user definitions in the form: `username:password`
- username is a user login name, and password is the DES-encrypted password.

Must not appear within a `<Limit>` or `<LimitExcept>` range.

Effect

Specifies that the named user file is to be used for any user names referenced in a require user or require valid-user directive.

Note that the use of `groups-with-users=yes` in the `Init fn=htaccess-init` directive in `obj.conf`, or specifying an `AuthGroupFile` directive with the same filename, causes that file to be assumed to be in the format:

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

AuthName

Syntax

`AuthName` authentication realm where authentication realm is a string identifying an authorization realm to be associated with any request for user authentication.

Must not appear within a `<Limit>` or `<LimitExcept>` range.

Effect

The authentication realm string typically appears in the prompt for username and password on the client side. It may affect caching of username and password on the client.

AuthType

Syntax

`AuthType` Basic. Must not appear within a `<Limit>` or `<LimitExcept>` range.

Effect

Specifies the user authentication method as HTTP Basic Authentication, the only method currently supported.

<Limit>

Syntax

```
<Limit method method ...>
```

```
allow, deny, order, or require directives
</Limit>
```

where method is an HTTP method such as GET, POST, or PUT. Any method that the web server understands can be used here.

Effect

Applies the enclosed directives only for requests using the specified HTTP methods.

<LimitExcept>

Syntax

```
<LimitExcept method method ...>
allow, deny, order, or require directives
</LimitExcept>
```

where method is an HTTP method such as GET, POST, or PUT. Any method that the web server understands can be used here.

Effect

Applies the enclosed directives only for requests types not matching the specified HTTP methods.

order

Syntax

Order ordering where ordering is one of:

- allow, deny
- deny, allow
- mutual-failure

Does not need to be enclosed within a `<Limit>` or `<LimitExcept>` range, but usually is.

Effect

- allows, denies, evaluates allow directives and then deny directives
- denies, allows, evaluates deny directives and then allow directives

- mutual-failure denies access for a host listed in both allow and deny directives, regardless of their ordering

require

Syntax

- require group groupname groupname
- require user username username
- require valid-user

Does not need to be enclosed within a `<Limit>` or `<LimitExcept>` range, but usually is.

Effect

- require group requires the authenticated user to be a member of one of the specified groups.
- require user requires the authenticated user to be one of the specified users.
- require valid-user requires an authenticated user

.htaccess Security Considerations

By default, server support for HTTP PUT is disabled. You can activate HTTP PUT using the Remote File Manipulation page of Content Mgmt in the Class Manager. Great care should be taken in allowing PUT access to directories containing `.htaccess` files, since it will allow them to be replaced. PUT access can be prevented on all files in a directory by restricting access. See "Restricting Access to a Directory (Path)" on page 209.

Controlling Access for Virtual Servers

Access control information in Sun ONE Web Server 6.1 can come from a per-virtual server ACL file and `.htaccess` files in the document directories. The `.htaccess` system is unchanged from iPlanet Web Server 4.x.

Your `server.xml` file can contain one or more ACLFILE tags which define an ID associated to a particular standard Sun ONE Web Server 6.x ACL file. For example:

```
<ACLFILE id="standard" file="standard.acl">
```

For virtual servers to use access control you must create a reference to one or more ACL file IDs in their ‘aclids’ property. Example:

```
<VS aclids="standard">
```

This configuration allows multiple virtual servers to share the same ACL file. If you want to require user-group authentication for a virtual server, you must add one or more USERDB tags to its definition. These USERDB tags create a connection between the database names in your ACL file and the actual databases found in `dbswitch.conf`.

The following example maps the ACLs with no ‘database’ attribute to the ‘default’ database in `dbswitch.conf`:

```
<VS>
   <USERDB id="default" database="default"/>
</VS>
```

Accessing Databases from Virtual Servers

You can globally define user authentication databases in the `dbswitch.conf` file. It is only read at server startup.

The `baseDN` of the LDAP URL in `dbswitch.conf` defines the global root of all accesses to the database. This maintains backward compatibility. For most new installations, the `baseDN` would be empty.

`dcsuffix` is a new attribute for LDAP databases in `dbswitch.conf` that defines the root of the DC tree according to the Sun ONE LDAP schema. It is relative to the `baseDN` in the LDAP URL. When the `dcsuffix` attribute is present, the LDAP database is Sun ONE LDAP schema compliant, and the behaviour of some operations changes. For more information about the Sun ONE LDAP schema, and an example, see “The Sun ONE LDAP Schema” in Chapter 2 of the Sun ONE Web Server 6.1 *Administrator’s Configuration Reference*.

For every virtual server, you can define one or more `USERDB` blocks that point to one of the directories, and you can define additional information. The `USERDB` blocks ID can be referenced in the database parameter of the ACL. If a virtual server has no `USERDB` blocks, user or group-based ACLs will fail.

`USERDB` tags define an additional layer of indirection between the database attribute of an ACL and `dbswitch.conf`. This layer of indirection adds the necessary protection for the server administrator to have full control over which databases virtual server administrators have access to.

For more information on USERDB, see "User Database Selection" in Chapter 2 of the Sun ONE Web Server 6.1 *Administrator's Configuration Reference.*

Specifying LDAP Databases in the User Interface

After you have defined one or more user authentication databases in `dbswitch.conf`, you can use the Class Manager to configure which databases each of your virtual servers will use for authentication. You can also use the Class Manager to add a newly created database definition from `dbswitch.conf` for the virtual server to authenticate against.

To specify which LDAP database or databases a virtual server should use, perform the following steps:

1. Access the Server Manager and select the Virtual Server Class tab.
2. Click on the virtual server class link where you wish to specify the LDAP database listed under Tree View of the Server.
3. Select the Virtual Servers tab, if not already displayed.
4. Click the ACL Settings link.

   The ACL Settings for Virtual Servers page is displayed.
5. Choose Edit from the drop-down list in the Option column, if not already displayed.
6. Select a database configuration from the drop-down list in the Database column of the virtual server you are editing.
7. Click OK.
8. Close the Edit ACL Files window.
9. Click Apply.
10. Choose dynamically apply.

Editing Access Control Lists for Virtual Servers

ACLs for virtual servers are created for the server instance that the virtual server resides in. Virtual server ACL settings default to those created for the server instance. However, access control for each virtual server can be edited through the Class Manager. You would also use this method to add a newly created ACL file to a virtual server.

To edit ACL settings for a virtual server, perform the following steps:

1. Access the Server Manager and select the Virtual Server Class tab.
2. Click on the virtual server class link where you wish to specify the LDAP database listed under Tree View of the Server.
3. Select the Virtual Servers tab, if not already displayed.
4. Click the ACL Settings link.
5. Choose Edit or Delete from the drop-down list in the Option field for each virtual server you wish to change.
6. Click the Edit link in the ACL File field to display the available ACL files.
7. Select one or more ACL files to add or delete for the virtual server.

   A virtual server can have multiple ACL files because they may have multiple document roots.
8. Choose the database to associate the ACL list with from the drop-down list.
9. (Optional) Enter the BaseDN.
10. Click OK when you have finished making changes.
11. Click Apply.
12. Select dynamically apply.

Creating ACLs For File-based Authentication

Sun ONE Web Server 6.1 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 Sun ONE Web Server 6.1 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.

An ACL entry can reference a user database using the `database` keyword. For example:

```
acl "default";
    authenticate (user) {
```

```
...
   database="myfile";
...
};
```

The database `myfile` can be referenced in the `USERDB` element of a `VS` in `server.xml` where it is linked with a corresponding definition in the `server-root/userdb/dbswitch.conf` file. Example:

```
<VS>
...
   <USERDB id="myfile" database="myfiledb">
...
</VS>
```

In the `server-root/userdb/dbswitch.conf` file there is an entry which defines the file auth-db and its configuration. Example:

```
directory myfiledb file
myfiledb:syntax keyfile
myfiledb:keyfile /path/to/config/keyfile
```

The table bow

Table 9-3 Parameters supported by the File Authentication Database

syntax	[Optional] Value is either keyfile, digest or htaccess. 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.
groupfile	[Required if syntax=htaccess] Path to the AuthGroupFile.
userfile	[Required if syntax=htaccess] Path to the AuthUserFile.

CAUTION The maximum length of a line in a file authentication database file (htaccess,digestfile or keyfile) is 255.

If any line exceeds this limit, the server will fail to start and an error will be logged in the log file.

NOTE Ensure that the following pre-conditions are met before you attempt to set ACLs using a file-based authentication database:

- A file-based authentication directory service is already configured. For information on how to do so, see "Configuring a Directory Service" on page 55.
- The virtual server on which the ACLs will be set is configured to use the type of file-based authentication database (keyfile, htaccess or digestauth) you require. If this is not done, ACL restrictions will be configured against the directory service configured as default.

Creating an ACL for a Directory Service Based on File Authentication

To create an ACL entry for a directory service based on file authentication, perform the following steps:

1. Access the Server Manager and select the server instance you wish to create or edit ACLs for.
2. Choose the Preferences tab from the Server Manager.
3. Click the Restrict Access link.
4. Under the Option column, choose the ACL file from the drop-down list and click Edit ACL.
5. In the Access Control Rules page in the top frame, click the Users / Groups link for the ACL you want to edit.
6. In the User / Group page in the bottom frame, from the Authentication database drop-down list, select keyfile.

7. Click Update.

   When you set an ACL against a keyfile-based file authentication database, the `dbswitch.conf` file is updated with an ACL entry, like the sample entry given below:

```
version 3.0;
  acl "default";
  authenticate (user) {
  prompt = "Sun One Web Server 6.1";
  database = "mykeyfile";
  method = "basic";
  };
deny (all) user = "anyone";
allow (all) user = "all";
```

Creating an ACL for a Directory Service Based on .htaccess Authentication

Sun ONE Web Server provides support for .htaccess-based flat file authentication. If you have been using .htaccess authentication, you can migrate your existing data files with no change to the file authentication database. As noted in Using .htaccess Files, .htaccess user and group data can be stored in a single file or split into two files (one with user data and other with group data). Both existing formats are supported by the file authentication database.

To create an ACL for a directory service based on htaccess authentication, perform the following steps:

1. Access the Server Manager and select the server instance you wish to create or edit ACLs for.
2. Choose the Preferences tab from the Server Manager.
3. Click the Restrict Access link.
4. Under the Option column, choose the ACL file from the drop-down list and click Edit ACL.
5. In the Access Control Rules page in the top frame, click the Users / Groups link for the ACL you want to edit.

6. In the User/Group page in the bottom frame, from the Authentication database drop-down list, select htaccess.
7. Click Update.

   When you set an ACL against an htaccess-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 One Web Server 6.1";
  database = "myhtaccessfile";
  method = "basic";
  };
deny (all) user = "anyone";
allow (all) user = "all";
```

Migrating Existing .htaccess information to the File Authentication Database

To migrate your existing .htaccess information to the file authentication database in Sun ONE Web Server 6.1:

- Copy your .htaccess userfile database to *server-root*`/`*server-instance*`/config/userfile`.
- Copy your htaccess groupfile database to *server-root*`/`*server-instance*`/config/groupfile`

The user file format is as follows:

```
#user:password
```

The group file format is as follows:

```
#group1:user1 user2
#group2:user3 user4
```

NOTE Member names are separated by spaces.

When userfile and groupfile have the same file name, they are combined. each line of the combination follows the syntax shown below:

```
#user:password:group1,group2
```

NOTE Columns are separated by colons.

Sample htaccess databases

Sample 1

```
#sample userfile  (user/password "j2ee/j2eepwd"  user/password
"user1/user1pwd" )

j2ee:9hmjfRwNxvJLU

user1:wvQirF86BsjSk
```

Sample 2

```
#sample group file

staff:j2ee  user1

eng:j2ee
```

Sample 3

```
#sample user/group file (username "j2ee", user password "j2eepwd")

j2ee:9hmjfRwNxvJLU:staff,eng
```

Creating an ACL for a Directory Service 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; clear text passwords are not maintained.

To create an ACL for a directory service based on digestauth-based authentication, perform the following steps:

1. Access the Server Manager and select the server instance you wish to create or edit ACLs for.
2. Choose the Preferences tab from the Server Manager.
3. Click the Restrict Access link.

4. Under the Option column, choose the ACL file from the drop-down list and click Edit ACL.
5. In the Access Control Rules page in the top frame, click the Users / Groups link for the ACL you want to edit.
6. In the User / Group page in the bottom frame, from the Authentication database drop-down list, select digest.
7. Click Update.

   When you set an ACL against a digestauth-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 10

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, quality of service features, or SNMP, see Monitoring Servers.

This chapter contains the following sections:

About Log Files

Server log files record your server's activity. You can use these logs to monitor your server and to help you when troubleshooting. The error log file, located in `https-`*`server_name`*`/logs/errors` in the server root directory, lists all the errors the server has encountered. The access log, located in `https-`*`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 Sun ONE Web Server access log file. You use the log analyzer to generate server statistics. You can back up server error and access log files by archiving them.

NOTE Due to limitations in the operating system, Sun ONE Web 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 the UNIX and Windows Platform

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 server `https-admserve/logs/` directory. Logs from the server instances are collected in the `https-`*`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.

Another feature available is to log the virtual server ID with the log message. This is a useful feature when multiple virtual servers are used to log messages to the same log file. You can choose to write the log messages to system log. When you do so, logging is not performed on the error log file. Instead the syslog logging service on UNIX, or the system logging service on Windows platform is used to produce and manage logs.

You can also use the `server.xml` attributes to control the contents of this file. For details about the `server.xml` file, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference.*

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 or virtual server 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 Sun ONE Web Server, and other daemon applications are collected in the same file, logged messages are enhanced with the following information to identify Sun ONE Web Server-specific messages from the particular server or virtual server instance:

- Unique message ID
- Timestamp

- Instancename
- Program name (webservd or webserv-wdog)
- Process ID (PID of the webserv 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 Sun ONE Web Server, in increasing order of severity.

Table 10-1 Log Levels

Log level	Description
`finest`	Messages indicate extent of verbosity of debug messages. `finest` gives the maximum verbosity.
`finer`	
`fine`	
`info`	Messages are informative in nature, usually related to server configuration or server status. These messages do not indicate errors that need immediate action.

Table 10-1 Log Levels

Log level	Description
`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.

About Virtual Servers and Logging

The Sun ONE Web Server can have virtual server instances. Each virtual server within a Sun ONE Web Server instance has its own identity and may have its own log file. The use of separate log files for each virtual server can help track server activity for particular transactions and resources.

You can also direct logged messages from multiple virtual servers to one server log file. When so doing, you may wish to enable the `logvsid` in the `LOG` element of the `server.xml` file. This helps users to distinguish log messages originating from different virtual servers.

```
<SERVER>

  ...

  <LOG file="/export//https-iws-files2.red.iplanet.com/logs/errors"
loglevel="finest" logtoconsole="true" usesyslog="false"
createconsole="false" logstderr="true" logstdout="true"
logvsid="true"/>

</SERVER>
```

In this example, `<LOG logvsid="true">` is responsible for including the virtual server ID in every log message. This allows you to differentiate messages coming from different virtual servers. The absence of attribute `errorlog` in the `VS` element, causes all the virtual servers to log messages to a single file.

Redirecting Application and Server Log Output

For developers, it is important that the application logs and server logs be made readily available during unit testing for Web application components and J2EE applications. On the Windows platform, developers prefer to see log messages displayed in a command window on the desktop. On the UNIX platform, many developers are comfortable with simply having the log messages stream to stderr in the terminal window in which the server instance is started, or, use the command `tail -f` to see the log messages written in log files.

The `server.xml` file contains attributes that can be set for stdout and stderr to direct logged messages to a log file or to the terminal window, and so forth. See the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference* for more information on the use of stdout and stderr.

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. Sun ONE Web Server saves the old log files and stamps the saved file with a name that includes the date and time they were saved.

For example, you can set up your access log files to rotate every hour, and Sun ONE Web Server saves and names the file "`access.200307152400`," 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.

Sun ONE Web Server offers the two types of log rotation for archiving files: Internal-daemon log rotation and Cron-based log rotation.

Internal-daemon Log Rotation

This type of log rotation happens within the HTTP daemon, and can only be configured at startup time. Internal daemon log rotation allows the server to rotate logs internally without requiring a server restart. Logs rotated using this method are saved in the following format:

`access.<YYYY><MM><DD><HHMM>`

`error. <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.200307152400`. 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 stored in the `scheduler.conf` file in the *`server_root`*`/https-admserv/config/` directory. This method allows you to archive log files immediately or have the server archive log files at a specific time on specific days. The server's scheduler configuration options are stored in `schedulerd.conf` in the *`server_root`*`/https-admserv/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.200307151630` when it is rotated at 4:30 p.m.

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

Once the rotation starts, Sun ONE Web 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 Files 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.

To add `%vsid%` to the log file format string:

1. Access the Server Manager and choose the Logs tab.
2. Click the Access Log Preferences link.
3. Enter a new log file location and filename in the Log File: text box.
4. Click the Only Log: radio button.
5. Click the Virtual Server Id check box. Alternatively to this, you can click the Custom Format: radio button and add the string '`%vsid%`.

NOTE When adding the custom format string '`%vsid%`, you must use a new access log file.

When changing the format of an existing log file, you should first delete / rename the existing log file OR use a different file name.

Server access logs can be in Common Logfile Format, flexible log format, or your own customizable format. The Common Logfile Format is a commonly supported format that provides a fixed amount of information about the server. The flexible log format allows you to choose (from Sun ONE Web Server) what to log. A customizable format uses parameter blocks that you specify to control what gets logged. For a list of customizable format parameters, see the *NSAPI Programmer's Guide*.

Once an access log for a resource has been created, you cannot change its format unless you archive it or create a new access log file for the resource.

You can specify logging preferences using the Access Log Preferences page in the Server Manager, or you can manually configure the following directives in the `obj.conf` file. In `magnus.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).

For more information on the NSAPI logging functions, including valid directives and parameters, see the *NSAPI Programmer's Guide.*

Easy Cookie Logging

Sun ONE Web Server has an easy way to log a specific cookie using the flexlog facility. Add "`Req->headers.cookie.`*`cookie_name`*" to the line that initializes the `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

Sun ONE web Server 6.1 allows you to configure the information to be logged in the server's errors logs.

For the Administration Server instance

1. Access the Administration Server
2. Select the Preferences tab.
3. Click the Access Logging Options link.
4. Enter the required information.
5. Click OK and then Apply to save and apply the changes.

For the Server Instance

1. Access the server instance.
2. Select the Logs tab.
3. Click the Error Log Preferences link.
4. Enter the required information.
5. Click OK and then Apply to save and 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 attributes

Attribute	Default	Description
`file`	`errors`	Specifies the file that stores messages from the default virtual server. Messages from other configured virtual servers also go here, unless the `errorlog` attribute is explicitly specified in the `VS` element.
`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`.
`logvsid`	`false`	(optional) If true, virtual server IDs are displayed in the virtual server logs. These are useful if multiple `VS` elements share the same log file. Note that in Sun ONE Web Server 6.1 the `logvsid` element cannot be configured in the `magnus.conf` file.
`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`.

Table 2 LOG attributes

Attribute	Default	Description
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 an Access Log File

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

To view the Administration Server's access log from the Administration Server, choose the Preferences tab, and then choose the View Access Log page.

To view an access log for the server instance from the Server Manager, choose the Logs tab, and then choose the View Access Log page

To view an access log for an individual virtual server from the Class Manager, select a virtual server to manage from the highlighted Manage Virtual Servers page, then click the link under the heading Access Log on the Virtual Server Manager page. You can specify the number of entries to view or entries with a conditional qualifier of your choice.

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 242 for more information):

```
wiley.a.com - - [16/Feb/2001:21:18:26 -0800] "GET / HTTP/1.0" 200 751
wiley.a.com - - [17/Feb/2001:1:04:38 -0800] "GET /docs/grafx/icon.gif
HTTP/1.0" 204 342
wiley.a.com - - [20/Feb/2001:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
arrow.a.com - john [29/Mar/2001:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
```

Table 10-3 describes the last line of this sample access log.

Table 10-3 The fields in the last line of the sample access log file

Access Log Field	Example
Hostname or IP address of client	arrow.a.com. (In this case, the hostname is shown because the web server's setting for DNS lookups is enabled; if DNS lookups were disabled, the client's IP address would appear.
RFC 931 information	- (RFC 931 identity not implemented)
Username	john (username entered by the client for authentication)
Date / time of request	29 / Mar / 1999:4:36:53 -0800
Request	GET / help
Protocol	HTTP / 1.0
Status code	401
Bytes transferred	571

The following is an example of an access log using the flexible logging format (you specify the format in the Log Preferences page; see "Setting Access Log Preferences" on page 242 for more information):

```
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET /index.htm HTTP/1.0" "GET"
"/?-" "HTTP/   1.0" 304 0 - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-"
"HTTP/1.0" 304 0    - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-"
"HTTP/1.0" 304 0    - Mozilla/2.0 (X11; I; IRIX 5.3 IP22)
```

Viewing the Error Log File

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

To view the Administration Server's error log file, from the Administration Server, choose the Preferences tab, and choose the View Error Log page.

To view a server instance's error log file, from the Server Manager, choose the Logs tab, and choose the View Error Log page.

To view an error log for an individual virtual server, from the Class Manager, select a virtual server to manage from the highlighted Manage Virtual Servers page, then click the link under the heading Error Log on the Virtual Server Manager page. You can specify the number of entries to view or entries with a conditional qualifier of your choice.

The following are two examples of entries in the error log; the first example shows an informational message indicating successful start up of the server, the second example indicates that the client `wiley.a.com` requested the file `report.html`, but the file wasn't in the primary document directory on the server.

```
[[22/Jan/2001:14:31:41] info (39700): successful server startup
[22/Jan/2001:14:31:41] info (39700): SunONE-WebServer/6.1 BB1-01/22/2001 01:45
[22/Jan/2001:14:31:42] warning (13751): for host wiley.a.com trying to GET
/report.html, send-file reports: can't find
/usr1/irenem/ES60-0424/docs/report.html (File not found)
```

Running 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-root*`/extras/flex_anlg` 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 whether you've selected common or flexible log file format.

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 Sun ONE Web Server or the command line. The log analyzer cannot generate statistics for virtual servers other than the default server. However, statistics can be viewed for each virtual server as described in "Viewing an Access Log File" on page 245.

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/https/lib:$LD_LIBRARY_PATH
```

AIX:

```
LIBPATH=server_root/bin/https/lib:$LIBPATH
```

HP-UX:

```
SHLIB_PATH=server_root/bin/https/lib:$SHLIB_PATH
```

Windows:

```
path=server_root\bin\https\bin;%path%
```

NOTE Before running the log analyzer, you should archive the server logs. For more information about archiving server logs, see "Archiving Log Files" on page 240.

To run the log analyzer from the Server Manager, follow these steps:

1. From the Server Manager, click the Logs tab.
2. Click Generate Report.
3. Fill in the fields.
4. Click OK.

   The report appears in a new window.

For more information, see the Generate Report Page in the online help.

To analyze access log files from the command line, run the tool, `flexanlg`, which is in the directory *`server-install`*`/extras/flex_anlg`.

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

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

The following describes the syntax.

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

Viewing Events (Windows)

In addition to logging errors to the server error log (see “Viewing the Error Log File” on page 246), Sun ONE Web Server logs severe system errors to the Event Viewer. The Event Viewer lets you monitor events on your system. Use the Event Viewer to see errors resulting from fundamental configuration problems, which can occur before the error log can be opened.

To use the Event Viewer, perform the following steps:

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 Sun ONE Web Server has a source label of `https-`*`serverid`* or `WebServer6.1`.
3. Choose Find from the View menu to search for one of these labels in the log. Choose Refresh from the View menu to see updated log entries.

   For more information about the Event Viewer, consult your system documentation.

Chapter 11

Monitoring Servers

This chapter contains information on ways to monitor your server, including the built-in monitoring tool, the quality of service features, and Simple Network Management Protocol (SNMP).

You can use SNMP together with Sun ONE 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 Sun ONE Web Server 6.1, ensure that Windows SNMP components are already installed on your machine.

You can view the server's status in real time by using the statistics feature or the SNMP. If you're using UNIX or Linux, you must configure your Sun ONE 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 Sun ONE server.

The following topics are included in this chapter:

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. You can view some statistics for individual virtual servers, and others for the entire server instance. 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. For more information, see the online Sun ONE Web Server 6.1 *Performance Tuning, Sizing and Scaling Guide.*

Once you enable statistics, you can view statistics in the following areas:

- connections
- DNS
- KeepAlive
- cache
- virtual servers

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.

CAUTION When you enable statistics / profiling, statistics information will be available to any user of your server. See the description of stats-xml in the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide* for more information.

Enabling Statistics

To enable statistics, follow these steps:

1. From the Server Manager, click the Monitor tab.
2. Click Monitor Current Activity

3. Click Yes to enable statistics.
4. Click OK.
5. Click Apply to apply your changes. You do not need to restart the server.

For more information on enabling statistics, see the online help.

Using Statistics

Once you've enabled statistics, you can get a variety of information on how your server instance and your virtual servers are running. The statistics are broken up into functional areas.

To access statistics, follow these steps:

1. From the Server Manager, click the Monitor tab.
2. Click Monitor Current Activity.
3. From the drop-down list, choose the poll interval.

   The poll interval is the number of seconds between updates of the statistics information displayed.
4. From the drop-down list, choose the kind of statistics you want displayed.
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 poll interval.

You can use the data you see in statistics to tune your server. For more information, see the online Sun ONE Web Server 6.1 *Performance Tuning, Sizing, and Scaling Guide*.

Using Quality of Service

Quality of Service refers to the performance limits you set for a server instance virtual server class, or virtual server. For example, if you are an ISP, you might want to charge different amounts of money for virtual servers depending on how much bandwidth you allow them. You can limit two areas: the amount of bandwidth and the number of connections.

You can enable these settings for the entire server or for a class of virtual servers in the Server Manager from the Monitor tab. However, you can override these server or class-level settings for an individual virtual server. For more information on setting quality of service limits for an individual server, see "Configuring Virtual Server Quality of Service Settings" on page 336.

Two settings govern how traffic is counted and how often the bandwidth is recomputed: the recompute interval and the metric interval. The recompute is how often (in milliseconds) the bandwidth is computed. The metric interval is the period of time for which data is used in traffic calculations.

This section includes the following topics:

Quality of Service Example

The following example shows how the quality of service information is collected and computed:

The server has metric interval of 30 seconds.

The server starts up at a time of 0 seconds.

At time 1 second, an HTTP connection generates 5000 bytes of traffic to/from the server.

No more connections are made after that. At 30 seconds, the total traffic for the last 30 seconds is 5000 bytes.

At 32 seconds, the traffic sample from 1 second is discarded, since it is older than the 30 seconds of the metric interval. The total traffic for the last 30 seconds is now 0.

The recompute interval works similarly. The server's recompute interval is 100ms.

Continuing with the example, the bandwidth gets recomputed periodically every 100 milliseconds. The calculation is based on the amount of traffic as well as the metric interval.

At time 0 seconds, the bandwidth is calculated for the first time. The total traffic is zero, divided by the metric interval of 30 seconds, gives a bandwidth of zero.

At 1 second, the bandwidth is calculated for the 10th time (1000 milliseconds / 100 milliseconds). The total traffic is 5000 bytes, which is divided by 30 seconds. The bandwidth is 5000 / 30 = 166 bytes per second.

At 30 seconds, the bandwidth is calculated for the 300th time. The total traffic is 5000 bytes, which is divided by 30 seconds. The bandwidth is 5000 / 30 = 166 bytes per second.

At 32 seconds, the bandwidth is computed again for the 320th time. The traffic is now 0 (since the one connection that generated traffic is too old to be counted), divided by 30, gives a bandwidth of 0 bytes / second.

Setting Up Quality of Service

To configure the quality of service settings for a server instance or a class of virtual servers, you need to configure the settings in through the user interface. To actually enforce your quality of service settings, you must also set up Server Application Functions (SAFs) in your obj.conf file.

To configure quality of service, follow these steps:

1. From the Server Manager, click the Monitor tab.
2. Click Quality of Service.

   A page appears listing general settings for quality of service, followed by a list containing the server instance as a whole and each class of virtual servers.
3. To enable quality of service as a whole, click Enable.

   By default quality of service is enabled. Enabling quality of service increases server overhead slightly.
4. Choose the Recompute Interval.

   The recompute interval is the number of milliseconds between each computation of the bandwidth for all servers, classes, and virtual servers. The default is 100 milliseconds.

5. Choose the Metric Interval.

   The metric interval is the interval in seconds during which the traffic is measured. The default is 30 seconds. All bandwidth measured during this time is averaged to give the bytes per second.

   If your site has a lot of large file transfers, use a large value (several minutes or more) or this field. A large file transfer might take up all the allowed bandwidth for a short metric interval, and result in connections being denied if you've enforced the maximum bandwidth setting. Since the bandwidth is averaged by the metric interval, a longer interval smooths out spikes caused by large files.

   If the bandwidth limit is much lower than available bandwidth (for example, 1 MB-per-second bandwidth limit but with a 1 GB-per-second connection to the backbone), the metric interval should be shortened.

   Please note that if you have large static file transfers and a bandwidth limit that is much lower than available bandwidth, you have to decide which situation to tune for, since the problems require opposite solutions.

6. Enable quality of service for the server instance and/or the virtual server classes.

   The lower portion of the screen lists the server instance and server classes. Choose Enable as the action next to the items for which you want to enable quality of service.

7. Set the maximum bandwidth, in bytes per second.

8. Choose whether or not to enforce the maximum bandwidth setting.

   If you choose to enforce the maximum bandwidth, once the server reaches its bandwidth limit additional connections are refused.

   If you do not enforce the maximum bandwidth, when the maximum is exceeded the server logs a message to the error log.

9. Choose the maximum number of connections allowed.

   This number is the number of concurrent requests processed.

10. Choose whether or not to enforce the maximum connections setting.

    If you choose to enforce the maximum connections, once the server reaches its limit additional connections are refused.

11. If you do not enforce the maximum connections, when the maximum is exceeded the server logs a message to the error log.

12. Click OK.

Required Changes to obj.conf

To enable quality of service, you must include directives in your `obj.conf` to invoke two Server Application Functions (SAFs): an AuthTrans `qos-handler` and an Error `qos-error`.

The `qos-handler` AuthTrans directive must be the first `AuthTrans` configured in the default object in order to work properly. The role of the quality of service handler is to examine the current statistics for the virtual server, virtual server class, and global server, and enforce the limits by returning an error.

Sun ONE Web Server includes a built-in sample quality of service handler SAF, called `qos-handler`. This SAF logs when limits are reached, and returns 503 "Server busy" to the server so that it can be processed by NSAPI.

Sun ONE Web Server also includes a built-in sample error SAF called `qos-error` which returns an error page stating which limits caused the 503 error and the value of the statistic that triggered the limit. You may want to alter the sample code to provide different error information.

These samples are available at *server_root*`/plugins/nsapi/examples/qos.c`. You can use these samples, or you can write your own SAFs.

For more information on these SAFs and how to use them, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide.*

Known Limitations to Quality of Service

When you use the quality of service features, keep in mind the following limitations:

- The connection or bandwidth statistics are not shared across server processes because of performance. In other words, the setting of `MaxProc` is not accounted for. So all the limits apply individually to a server process, not to the aggregate of all processes. For more information on `MaxProcs` and multiple processes, see the online Sun ONE Web Server 6.1 *Performance Tuning, Sizing, and Scaling Guide.*
- The quality of service features only measure the HTTP bandwidth at the application level. The HTTP bandwidth can differ from the actual TCP network bandwidth for a variety of reasons:

- If SSL is enabled, handshakes and client certificate exchanges add to the traffic but are not measured.
- If chunked encoding is enabled in either or both directions, the chunking layer removes the chunk headers and they are not counted in the traffic. Other headers or protocol items are counted.

- The quality of service features cannot accurately measure traffic from `PR_TransmitFile` calls. For basic I/O operations such as `PR_Send()/net_write` or `PR_Recv()/net_read`, the data transferred can be quickly accounted for by the bandwidth manager, since the number of bytes transferred in one system call is usually the size of a buffer and the I/O call returns quickly. This works very well to measure the instantaneous bandwidth of dynamic content applications. However, because the amount of data transferred from `PR_TransmitFile` is only known at the end of the transfer, it can't be measured before it completes.

  If the `PR_TransmitFile` is short, then the quality of service features will perform adequately. However, If the `PR_TransmitFile` is long, such as in the case of a long file downloaded by a dialup user, the whole amount of data transferred will be counted at completion time. When the bandwidth manager recomputes bandwidth after the next recompute interval period starts, the bandwidth computed will go up significantly because of that recent large `PR_TransmitFile`. This case could cause the server to deny all requests until the next metric interval, when the bandwidth manager will "expire" the transmit file operation, since it is too old, and thus the bandwidth value will go back down. If your site has a lot of very long static file downloads, the you should increase the metric interval from the default 30 seconds.

- The bandwidth computed is always an approximation because it is not measured instantaneously, but is recomputed at regular intervals and over a certain period. For example, if the metric interval is the default 30 seconds and the server is idle for 29 seconds, then the next second, a client could potentially use 30 times the bandwidth limit in one second.

- The quality of service bandwidth statistics are lost whenever the server is reconfigured dynamically. In addition, the quality of service limitations are not enforced in threads that have connections on an older, inactive configuration, because the bandwidth manager thread only computes bandwidth statistics for the active configuration. Potentially, a client that doesn't close its socket for a long time and remains active so that the server doesn't time it out would not be subject to the quality of service limitations after a server dynamic reconfiguration.

- The concurrent connections are computed with a different granularity for virtual servers than for virtual server classes and the global server instance. The connection counter for an individual virtual server is incremented atomically immediately after the request is parsed and routed to the virtual server. It is also decremented atomically at the end of the response processing for that request. This means that the virtual server connection statistics are always exact at any instant.

  However, the connection statistics for the virtual server class and global server instance are not updated instantly. They are updated by the bandwidth manager thread every recompute interval. The connection count for the virtual server class is the sum of the connections on all virtual servers of that class; and the global server instance connection count is the sum of connections on all virtual server classes.

  Because of the way these values are computed, the number of connections for a virtual server is always correct (and if you've enforced a limit to the number of connections, you can never have more than the limit), and the virtual server class and server instance values are not quite as accurate, since they're only computed at intervals.

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 web server, and other servers on your network. The NMS is a machine 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 Sun ONE 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.

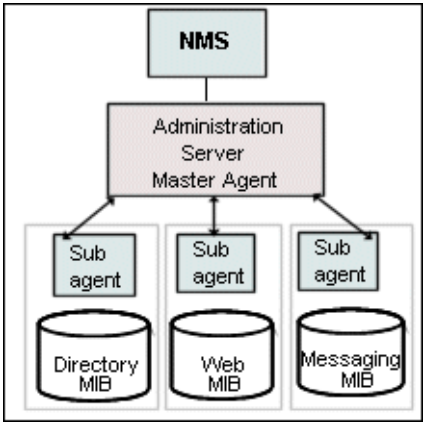
The subagent gathers information about the server and passes the information to the server's master agent. Every Sun ONE server, except for the Administration Server, has as subagent.

NOTE After making any SNMP configuration changes, you must click the Apply button, 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, Sun ONE Web Server, and the Messaging Server installed on the same host, the subagents for each of the servers would communicate with the same master agent, as shown below:

The Network Management Station and SNMP Agents

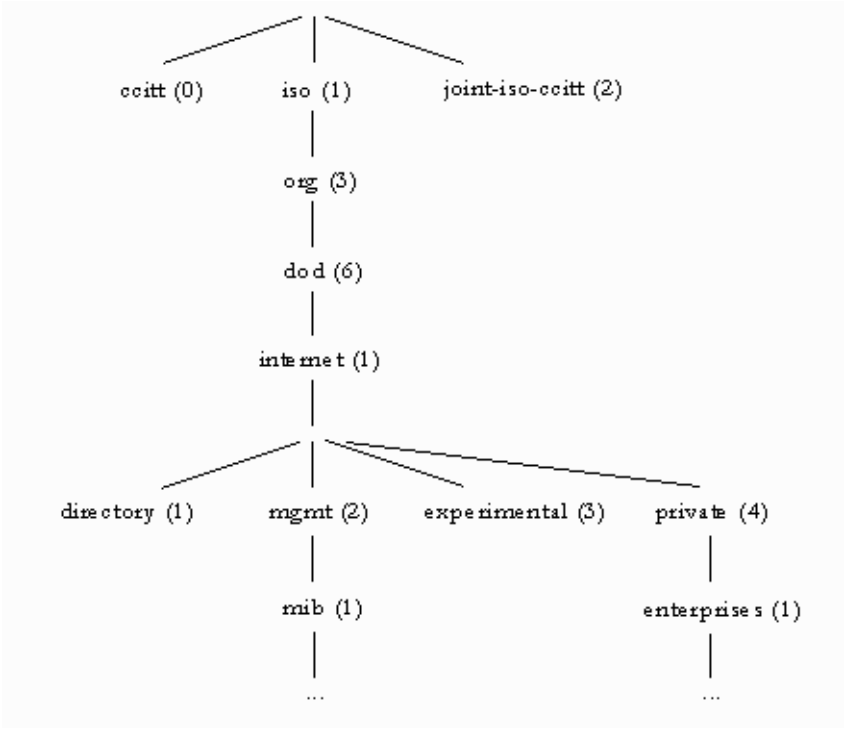


The Sun ONE Web Server MIB

Sun ONE Web 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 web server's network configuration, status, and statistics. Using SNMP, you can view this information from the network management workstation (NMS).

A server's MIB contains variable definitions pertaining to network management for that particular server. The top level of the MIB tree is shown in the figure below:

Top level of the MIB tree



The top level of the MIB tree shows that the internet object identifier has four subtrees: directory (1), mgmt (2), experimental (3), and private (4). The private (4) subtree contains the enterprises (1) node. Each subtree in the enterprises (1) node is assigned to an individual enterprise, which is an organization that has registered its own specific MIB extensions. An enterprise can then create product-specific subtrees under its subtree. MIBs created by companies are located under the enterprises (1) node. The Sun ONE MIBs are located under the enterprises (1) node.

Each Sun ONE server subagent provides a MIB for use in SNMP communication. The server reports significant events to the network management station (NMS) by sending messages or traps containing these variables. The NMS can also query the server's MIB for data, or can remotely change variables in the MIB.

Each Sun ONE server has its own management information base (MIB). All Sun ONE MIBs are located at:

server_root`/plugins/snmp`

The Sun ONE Web Server's MIB is a file called `webserv61.mib`. This MIB contains the definitions for various variables pertaining to network management for Sun ONE Web Server.

The Sun ONE Web Server 6.1 MIB has an object identifier of `http 60 (iws60 OBJECT IDENTIFIER ::= {http 60 })` and is located in the *server_root*`/plugins/snmp` directory.

You can see administrative information about your web server and monitor the server in real time using the Sun ONE Web Server MIB. Table 11-1 lists and describes the managed objects stored in the `webserv61.mib`.

Table 11-1 `webserv61.mib` managed objects and descriptions

Managed object	Description
iwsInstanceTable	Sun ONE Web Server instances.
iwsInstanceEntry	Sun ONE Web Server instance.
iwsInstanceIndex	Server instance index.
iwsInstanceId	Server instance identifier
iwsInstanceVersion	String, such as `SunONE-WebServer/6.1 BB1-01/24/2001 17:15 (SunOS DOMESTIC)`
iwsInstanceDescription	Description of the server instance.
iwsInstanceOrganization	Organization responsible for the server instance.
iwsInstanceContact	Contact information for person(s) responsible for server instance.
iwsInstanceLocation	Where the server is located.
iwsInstanceStatus	Status of the server instance.
iwsInstanceUptime	How long the server has been running.
iwsInstanceDeathCount	Number of times server instance processes have gone down.
iwsInstanceRequests	Number of requests processed by the server instance.
iwsInstanceInOctets	Number of octets received by the server instance. Will show 0 if information is not available.

Table 11-1 `webserv61.mib` managed objects and descriptions *(Continued)*

Managed object	Description
iwsInstanceOutOctets	Number of octets transmitted by the server instance. Will show 0 if information is not available.
iwsInstanceCount2xx	Number of 200-level (Successful) responses issued by the server instance.
iwsInstanceCount3xx	Number of 300-level (Redirection) responses issued by the server instance.
iwsInstanceCount4xx	Number of 400-level (Client Error) responses issued by the server instance.
iwsInstanceCount5xx	Number of 500-level (Server Error) responses issued by the server instance.
iwsInstanceCountOther	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued by the server instance.
iwsInstanceCount200	Number of 200 (Request Fulfilled) responses issued by the server instance.
iwsInstanceCount302	Number of 302 (Moved Temporarily) responses issued by the server instance.
iwsInstanceCount304	Number of 304 (Not Modified) responses issued by the server instance.
iwsInstanceCount400	Number of 400 (Bad Request) responses issued by the server instance.
iwsInstanceCount401	Number of 401 (Unauthorized) responses issued by the server instance.
iwsInstanceCount403	Number of 403 (Forbidden) responses issued by the server instance.
iwsInstanceCount404	Number of 404 (Not Found) responses issued by the server instance.
iwsInstanceCount503	Number of 503 (Unavailable) responses issued.
iwsVsTable	Sun ONE Web Server virtual servers.
iwsVsEntry	Sun ONE Web Server virtual server.
iwsVsIndex	Virtual server index.
iwsVsId	Virtual server identifier.

Table 11-1 `webserv61.mib` managed objects and descriptions *(Continued)*

Managed object	Description
iwsVsRequests	Number of requests processed by the virtual server.
iwsVsInOctets	Number of octets received by the virtual server.
iwsVsOutOctets	Number of octets transmitted by the virtual server.
iwsVsCount2xx	Number of 200-level (Successful) responses issued by the virtual server.
iwsVsCount3xx	Number of 300-level (Redirection) responses issued by the virtual server.
iwsVsCount4xx	Number of 400-level (Client Error) responses issued by the virtual server.
iwsVsCount5xx	Number of 500-level (Server Error) responses issued by the virtual server.
iwsVsCountOther	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued by the virtual server.
iwsVsCount200	Number of 200 (Request Fulfilled) responses issued by the virtual server.
iwsVsCount302	Number of 302 (Moved Temporarily) responses issued by the virtual server.
iwsVsCount304	Number of 304 (Not Modified) responses issued by the virtual server.
iwsVsCount400	Number of 400 (Bad Request) responses issued by the virtual server.
iwsVsCount401	Number of 401 (Unauthorized) responses issued by the virtual server.
iwsVsCount403	Number of 403 (Forbidden) responses issued by the virtual server.
iwsVsCount404	Number of 404 (Not Found) responses issued by the virtual server.
iwsVsCount503	Number of 503 (Unavailable) responses issued.
iwsProcessTable	Sun ONE Web Server processes.
iwsProcessEntry	Sun ONE Web Server process.

Table 11-1 `webserv61.mib` managed objects and descriptions *(Continued)*

Managed object	Description
iwsProcessIndex	Process index.
iwsProcessId	Operating system process identifier.
iwsProcessThreadCount	Number of request processing threads.
iwsProcessThreadIdle	Number of request processing threads currently idle.
iwsProcessConnectionQueueCount	Number of connections currently in connection queue.
iwsProcessConnectionQueuePeak	Largest number of connections that have been queued simultaneously.
iwsProcessConnectionQueueMax	Maximum number of connections allowed in connection queue.
iwsProcessConnectionQueueTotal	Number of connections that have been accepted.
iwsProcessConnectionQueueOverflows	Number of connections rejected due to connection queue overflow.
iwsProcessKeepaliveCount	Number of connections currently in keepalive queue.
iwsProcessKeepaliveMax	Maximum number of connections allowed in keepalive queue.
iwsProcessSizeResident	Process resident size in kbytes.
iwsProcessSizeVirtual	Process size in kbytes.
iwsProcessFractionSystemMemoryUsage	Fraction of process memory in system memory.
iwsListenTable	Sun ONE Web Server listen sockets.
iwsListenEntry	Sun ONE Web Server listen socket.
iwsListenIndex	Listen socket index.
iwsListenId	Listen socket identifier.
iwsListenAddress	Address where socket listens.
iwsListenPort	Port where socket listens.
iwsListenSecurity	Encryption support.
iwsThreadPoolTable	Sun ONE Web Server thread pools.
iwsThreadPoolEntry	Sun ONE Web Server thread pool.

Table 11-1 `webserv61.mib` managed objects and descriptions *(Continued)*

Managed object	Description
iwsThreadPoolIndex	Thread pool index.
iwsThreadPoolID	Thread pool identifier.
iwsThreadPoolCount	Number of requests queued.
iwsThreadPoolPeak	Largest number of requests that have been queued simultaneously.
iwsThreadPoolMax	Maximum number of requests allowed in queue.
iwsInstanceStatusChange	An `iwsInstanceStatusChange` trap signifies that `iwsInstanceStatus` has changed.
iwsInstanceLoad1MinuteAverage	System load average for one minute.
iwsInstanceLoad5MinuteAverage	System load average for five minutes.
iwsInstanceLoad15MinuteAverage	System load average for fifteen minutes.
iwsInstanceNetworkInOctets	Number of octets transmitted on the network per second.
iwsInstanceNetworkOutOctets	Number of octets received on the network per second.
iwsCpuIndex	The CPU index.
iwsCpuId	The CPU id.
iwsCpuIdleTime	The CPU idle time.
iwsCpuUserTime	The CPU user time.
iwsCpuKernelTime	The CPU kernel time.

Setting Up SNMP

In general, to use SNMP you must have a master agent and at least one subagent installed and running on a your system. You need to install the master agent before you can enable a subagent.

The procedures for setting up SNMP are different depending upon your system. Table 8.1 provides an overview of procedures you will follow for different situations. The actual procedures are described in detail later in the chapter.

Before you begin, you should verify two things:

- Is your system already running an SNMP agent (an agent native to your operating system)?
- If so, does your native SNMP agent support SMUX communication? (If you're using the AIX platform, your system supports SMUX.)

See your system documentation for information on how to verify this information.

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 machine.
- (UNIX) Restart the SNMP master agent using the Administration Server.

Table 2 Overview of procedures for enabling SNMP master agents and subagents.

If your server meets these conditions....	...follow these procedures. These are discussed in detail in the following sections.
• No native agent is currently running	1. Start the master agent. 2. Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • No need to continue using native agent	1. Stop the native agent when you install the master agent for your Administration Server. 2. Start the master agent. 3. Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • Needs to continue using native agent	1. Install a proxy SNMP agent. 2. Start the 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.

If your server meets these conditions....	...follow these procedures. These are discussed in detail in the following sections.
• Native agent is currently running • SMUX supported	1. Reconfigure the SNMP native agent. 2. Enable the subagent for each server installed on the system.

Using a Proxy SNMP Agent (UNIX/Linux)

You need to use a proxy SNMP agent when you already have a native agent running, and you want to use continue using it concurrently with an Sun ONE Web 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'll need to install it and then start it. You'll also have to restart the native SNMP master agent using a port number other than the one the Sun ONE Web 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 270.
2. Install and start the proxy SNMP agent and restart the native SNMP daemon. See "Using a Proxy SNMP Agent (UNIX/Linux)" on page 268.
3. Start the SNMP master agent. See "Enabling and Starting the SNMP Master Agent" on page 271.
4. Enable the subagent. See "Enabling the Subagent" on page 276.

To install the SNMP proxy agent, edit the `CONFIG` file (you can give this file a different name), located in `plugins/snmp/sagt` in the server root directory, so that it includes the port that the SNMP daemon will listen to. It also needs to include the MIB trees and traps that the proxy SNMP agent will forward.

Here is an example 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'd enter:

```
# snmpd -P 1161
```

Reconfiguring the SNMP Native Agent

If your SNMP daemon is running on AIX, it supports SMUX. For this reason, you don't need to install a master agent. However, you do need to change the AIX SNMP daemon configuration.

AIX uses several configuration files to screen its communications. One of them, `snmpd.conf`, needs to be changed so that the SNMP daemon accepts the incoming messages from the SMUX subagent. For more information, see the online manual page for `snmpd.conf`. You need to add a line to define each subagent.

For example, you might add this line to the `snmpd.conf`:

```
smux 1.3.6.1.4.1.1.1450.1 "" IP_address net_mask
```

IP_address is the IP address of the host the subagent is running on, and *net_mask* is the network mask of that host.

NOTE Do not use the loopback address 127.0.0.1; use the real IP address instead.

Installing the SNMP Master Agent

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 using the Server Manager:

1. Log in as root.
2. Check whether an SNMP daemon (`snmpd`) is running on port 161.

   If no SNMP daemon is running, go to Step 4.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports.
3. If an SNMP daemon is running, kill its process.
4. In the Server Manager, choose the SNMP Master Agent Trap page from the Global Settings tab. The Manager Entries page appears.
5. Type the name of the system that is running your network management software.

6. Type the port number at which your network management system listens for traps. (The well-known port is 162.) For more information on traps, see “Configuring Trap Destinations” on page 276.
7. Type the community string you want to use in the trap. For more information on community strings, see “Configuring the Community String” on page 276.
8. Click OK.
9. In the Server Manager, the SNMP Master Agent Community page from the choose Global Settings tab. The Community Strings page appears.
10. Type the community string for the master agent.
11. Choose an operation for the community.
12. Click OK.

Enabling and Starting the SNMP Master Agent

Master agent operation is defined in an agent configuration file named `CONFIG`. You can edit the `CONFIG` file using the Server Manager, or you can edit the file manually. You must install the master SNMP agent before you can enable the SNMP subagent.

If you get a bind error similar to “System Error: Could not bind to port,” when restarting the master agent, use `ps -ef | grep snmp` to check if `magt` is running. If it is running, use the command `kill -9` *`pid`* to end the process. The CGIs for SNMP will then start working again.

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 Master Agent
- 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. However, you can manually start the master agent on another port using the following steps:

1. Edit */server_root*/plugins/snmp/magt/CONFIG to specify the desired port.
2. Run the start script as follows:

   ```
   cd /server_root/https-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

The CONFIG file defines the community and the manager that master agent will work with. The manager value should be a valid system name or an IP address.

Here is an example of a basic CONFIG file:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            manager_station_name
                   SEND ALL TRAPS TO PORT 162
                    WITH COMMUNITY public
```

Defining sysContact and sysLocation Variables

You can edit the `CONFIG` file to add initial values for `sysContact` and `sysLocation` which specify the `sysContact` and `sysLocation` MIB-II variables. The strings for `sysContact` and `sysLocation` in this example are enclosed in quotes. Any string that contains spaces, line breaks, tabs, and so on must be in quotes. You can also specify the value in hexadecimal notation.

Here is an example of a `CONFIG` file with `sysContract` and `sysLocation` variables defined:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            nms2
                   SEND ALL TRAPS TO PORT 162
                   WITH COMMUNITY public

INITIAL            sysLocation “Server room
501 East Middlefield Road
Mountain View, CA 94043
USA”

INITIAL            sysContact “John Doe
email: jdoe@netscape.com”
```

Configuring the SNMP Subagent

You can configure the SNMP subagent to monitor your server.

To configure the SNMP subagent, perform the following steps:

1. From the Administration Server, select the server instance and click Manage.
2. Select the Monitor tab.
3. Select SNMP Subagent Configuration.
4. (UNIX only) Enter the name and domain of the server in the Master Host field.
5. Enter the Description of the server, including operating system information.
6. Enter the Organization responsible for the server.
7. Enter the absolute path for the server in the Location field.
8. Enter the name of the person responsible for the server and the person's contact information in the Contact field.
9. Select On to Enable the SNMP Statistics Collection.
10. Click OK.
11. Click Apply.
12. Select Apply Changes to restart your server for changes to take effect.

Starting the SNMP Master Agent

Once you have installed the SNMP master agent, you can start it manually or by using the Administration Server.

Manually Starting the SNMP Master Agent

To start the master agent manually, enter the following at the command prompt:

```
# magt CONFIG INIT&
```

The `INIT` file is a nonvolatile file that contains information from the MIB-II system group, including system location and contact information. If `INIT` doesn't already exist, starting the master agent for the first time will create it. An invalid manager name in the `CONFIG` file will cause the master agent start-up to fail.

To start a master agent on a nonstandard port, use one of two methods:

Method one: In the `CONFIG` file, specify a transport mapping for each interface over which the master agent listens for SNMP requests from managers. Transport mappings allow the master agent to accept connections at the standard port and at a nonstandard port. The master agent can also accept SNMP traffic at a nonstandard port. The maximum number of concurrent SNMP is limited by your target system's limits on the number of open sockets or file descriptors per process. Here is an example of a transport mapping entry:

```
TRANSPORT          extraordinary   SNMP
                   OVER UDP SOCKET
                    AT PORT 11161
```

After editing the `CONFIG` file manually, you should start the master agent manually by typing the following at the command prompt:

```
# magt CONFIG INIT&
```

Method two: Edit the `/etc/services` file to allow the master agent to accept connections at the standard port as well as a nonstandard port.

Starting the SNMP Master Agent Using the Administration Server

To start the SNMP master agent using the Administration Server, perform the following steps:

1. Log in to the Administration Server.
2. In the Server Manager, choose the SNMP Master Agent Control page from the Global Settings tab. The SNMP Master Agent Control page appears.
3. Click Start.

   You can also stop and restart the SNMP master agent from the SNMP Master Agent Control page.

Configuring the SNMP Master Agent

Once you've enabled the master agent and enabled a subagent on a host computer, you need to configure the host's Administration Server. This entails specifying community strings and trap destinations.

Configuring the Community String

A community string is a text string that an SNMP agent uses for authorization. This means that a network management station would send a community string with each message it sends to the agent. The agent can then verify whether the network management station is authorized to get information. Community strings are not concealed when sent in SNMP packets; strings are sent in ASCII text.

You can configure the community string for the SNMP master agent from the Community Strings page in the Server Manager. You also define which SNMP-related operations a particular community can perform. From the Server Manager, you can also view, edit, and remove the communities you have already configured.

Configuring Trap Destinations

An SNMP trap is a message the SNMP agent sends to a network management station. For example, an SNMP agent sends a trap when an interface's status has changed from up to down. The SNMP agent must know the address of the network management station so it knows where to send traps. You can configure this trap destination for the SNMP master agent from Sun ONE Web Server. You can also view, edit, and remove the trap destinations you have already configured. When you configure trap destinations using Sun ONE Web Server, you are actually editing the `CONFIG` file.

Enabling the Subagent

After you have installed the master agent that comes with the Administration Server, you must enable the subagent for your server instance before you attempt to start it. For more information on installing the master agent, see "Installing the SNMP Master Agent" on page 270. You can use the Server Manager to enable the subagent.

To stop the SNMP function on UNIX/Linux platforms, you must stop the subagent first, then the master agent. If you stop the master agent first, you may not be able to stop the subagent. If that happens, restart the master agent, stop the subagent, then stop the master agent.

To enable the SNMP subagent, use the SNMP Subagent Configuration page in the Server Manager, and start the subagent from the SNMP Subagent Control page. For more information, see the corresponding sections in the online help.

Once you have enabled the subagent, you can start, stop or restart it from the SNMP Subagent Control page or the Services Control Panel for Windows.

NOTE After making any SNMP configuration changes, you must click the Apply button, 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.

Chapter 12

Configuring Naming and Resources

The component-based Java™ 2 Platform, Enterprise Edition (J2EE™) technology provides an infrastructure for Web services that simplifies enterprise development and deployment.

This chapter describes the J2EE resources provided by Sun ONE Web Server and discusses the methods used to create and manage these resources.

For a discussion on Java security and realm-based authentication, see Chapter 4, "J2EE-based Security for Web Container and Web Applications".

This chapter includes the following topics:

Enabling and Disabling Java

You can enable or disable Java either globally, that is, per instance of Sun ONE Web Server, or for a particular virtual server class. By default, Java is enabled in Sun ONE Web Server and the following line is added to the `magnus.conf` file:

```
Init fn="load-modules"
shlib="<server-root>/bin/https/lib/libj2eeplugin.so"
```

You can also enable Java for a particular virtual server. When you do so, the server updates the `obj.conf` file for the virtual server class with the required J2EE directives.

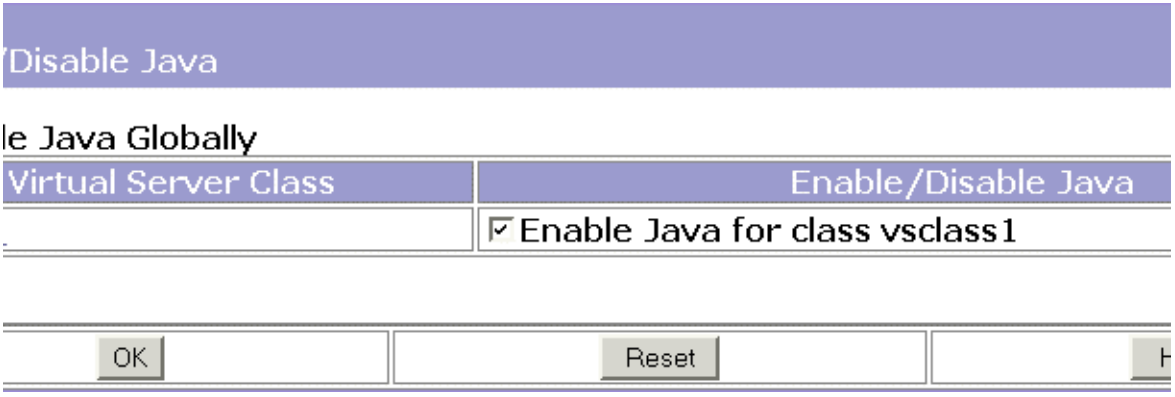
For more information on the `obj.conf` and `magnus.conf` files, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference* and the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide*.

In some cases, you might want to disable Java either globally or for a particular virtual server class, for example, if your entire server or that class will supply only static content.

To enable or disable Java, do the following:

1. Access the Server Manager and choose the Java tab.
2. Click Enable / Disable Servlets / JSP.

   The Enable / Disable Servlets / JSP Interface



3. To enable or disable Java globally, check or uncheck Enable / Disable Java Globally.

   or

   To enable or disable Java for a particular virtual server class, check or uncheck the Enable / Disable Java checkbox corresponding to the virtual server class.
4. Click OK.

Configuring JVM Settings

Unlike previous releases of the product, the Sun ONE Web Server 6.1 no longer supports the standalone Java Runtime Environment (JRE). Instead, JDK 1.4.1 or higher is pre-requisite for the server. When you install the server, if you select the default JDK option ,Java Development Kit (JDK) version 1.4.1_03 is installed in the `<server-root>/bin/https/jdk` directory.

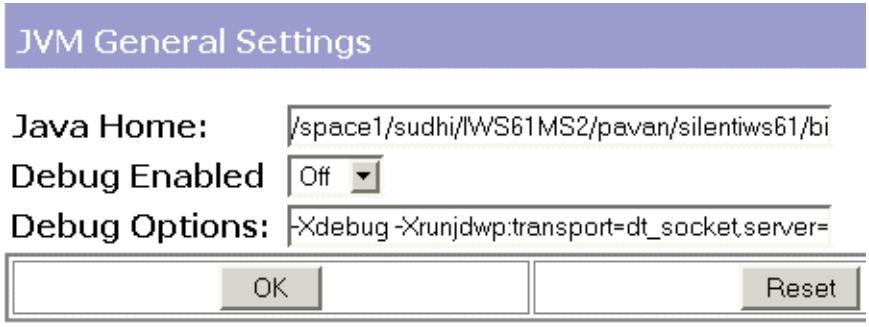
You can configure Java Virtual Machine (JVM) settings for your server instance. These settings include the location of your Java home, compiler options, debugging options, and profiler information. One reason to configure these settings is to improve performance. For more information on performance see the Sun ONE Web Server 6.1 *Performance Tuning, Sizing, and Scaling Guide.*

Configuring General Settings

To edit the location of the JDK and to specify debug options:

1. Access the Server Manager and choose the Java tab.
2. Click JVM General.

   The JVM General Interface



3. Set the Java Home.

   The Java Home is the path to the directory where the Java Developer's Kit (JDK) is installed. Sun ONE Web Server supports the Sun JDK 1.4.1_03.

4. Choose whether to enable debugging and set debug options.

   A list of debug options is available at:
   `http://java.sun.com/products/jpda/doc/conninv.html#Invocation`

5. Click OK.

Configuring Path Settings

You might want to configure JVM path settings for certain reasons. For example, you might want to choose a suffix for the system's classpath in order to override system classes such as the XML Parser classes, or you might want to ignore the environment classpath to prevent environment variable side effects on a production environment.

To configure the JVM's path settings in the Administration interface, perform the following tasks:

1. Access the Server Manager and choose the Java tab.
2. Click JVM Path Settings.
3. Choose a suffix for the system's classpath
4. Choose whether to ignore the environment classpath.

   If you do not ignore the classpath, the `CLASSPATH` environment variable is read and appended to the Sun ONE Web Server classpath. The `CLASSPATH` environment variable is added after the `classpathsuffix`, at the very end.

   For a development environment, the classpath should be used. For a production environment, this classpath should be ignored to prevent environment variable side effects.

5. Set a native library path prefix and suffix.

   The native library path is the automatically constructed concatenation of the Web Server installation relative path for its native shared libraries, the standard JRE native library path, the shell environment setting (`LD_LIBRARY_PATH` on UNIX), and any path specified in the `profiler` element. Since this is synthesized, it does not appear explicitly in the server configuration.

6. Click OK.

Configuring JVM Options

To set JVM command-line options in the Administration interface, perform the following tasks:

1. Access the Server Manager and choose the Java tab.
2. Click JVM Options and make the necessary changes.

   For information about specific JVM options, see:

   `http://java.sun.com/docs/hotspot/VMOptions.html`
3. Click OK.

Configuring the JVM Profiler

You can use a profiler to perform remote profiling on the Sun ONE Web Server to discover bottlenecks in server-side performance.

To configure the JVM Profiler in the Administration interface, perform the following tasks:

1. Access the Server Manager and choose the Java tab.
2. Click JVM Profiler.
3. Specify the classpath, the native library path, and whether the profiler is enabled.
4. Add, delete or edit JVM options for the profiler and click OK.

For more information about profilers, see the Sun ONE Web Server 6.1 *Programmer's Guide*.

About J2EE Naming Services and Resources

Web applications may access a wide variety of resources such as resource managers, data sources (for example SQL datasources), mail sessions, and URL connection factories. The J2EE platform exposes such resources to the applications via Java Naming and Directory Interface (JNDI) service.

Sun ONE Web Server allows you to create and manage the following J2EE resources:

- JDBC Datasources
- JDBC Connection Pools
- Java Mail Sessions

- Custom Resources
- External JNDI Resources

JDBC Datasources

A JDBC Datasource is a J2EE resource that you can create and manage using Sun ONE Web Server.

The JDBC API is the API for connectivity with relational database systems. The JDBC API has two parts:

- An application-level interface used by the application components to access databases.
- A service provider interface to attach a JDBC driver to the J2EE platform.

A JDBC Datasource object is an implementation of a data source in the Java programming language. In basic terms, a data source is a facility for storing data. It can be as sophisticated as a complex database for a large corporation or as simple as a file with rows and columns. A JDBC Datasource is a J2EE resource that can be created and managed via Sun ONE Web Server.

The JDBC API provides a set of classes for Java with a standard SQL database access interface to ensure uniform access to a wide range of relational databases.

Using JDBC, SQL statements can be sent to virtually any database management system (DBMS). It is used as an interface for both relational and object DBMSs.

For information on creating a custom resource, see Creating a JDBC Resource.

JDBC Connection Pools

A JDBC connection pool is a named group of JDBC connections to a database. These connections are created when the first request for connection is made on the pool when you start Sun ONE Web Server.

The JDBC connection pool defines the properties used to create a connection pool. Each connection pool uses a JDBC driver to establish a connection to a physical database at server start-up.

A JDBC-based application or resource draws a connection from the pool, uses it, and when no longer needed, returns it to the connection pool by closing the connection. If two or more JDBC resources point to the same pool definition, they will be using the same pool of connections at run time.

For information on how to create a new JDBC connection pool, see Creating a New JDBC Connection Pool.

Java Mail Sessions

JMS destinations are J2EE resources that can be created and managed via Sun ONE Web Server.

Many internet applications require the ability to send email notifications, so the J2EE platform includes the JavaMail API along with a JavaMail service provider that allows an application component to send internet mail. The JavaMail API has two parts:

- An application-level interface used by the application components to send mail
- A service provider interface used at the J2EE API level.

Java Mail Sessions are J2EE resources that can be created and managed via Sun ONE Web Server.

NOTE Sun ONE Web Server does not provide an Administration Server interface to create Java Mail Sessions. You can use the command line interface to do so. For more information on how you can create a mail resource using the command line utility, see Create Mail Resource.

Custom Resources

A custom resource accesses a local JNDI repository. The `customresource` element defined in `server.xml` provides a way of specifying a custom server-wide resource object factory. Such object factories implement the `javax.naming.spi.ObjectFactory` interface. This element associates a JNDI name (specified through the `jndiname` sub-element like other Sun ONE Web Server resources) to be used in the server-wide namespace, its type, name of the resource factory class and a set of standard properties used to instantiate the same.

You need to ensure that the resource reference's environment references are linked to the configured server-wide resources defined using the `customresource` and `externaljndiresource` tags in `server.xml`. Dynamic redeployment of application components is an issue for the JNDI naming environment. Sun ONE Web Server will release all the application specific references and rebind all the new references into the newly installed application's naming context.

For information on creating a custom resource, see Creating Custom Resources.

External JNDI Resources

Often applications running on Sun ONE Web Server require access to resources stored in an external JNDI repository. For example, generic Java objects could be stored in an LDAP server as per the Java schema. While a custom resource allows you to access a local JNDI repository, to access an external JNDI repository you must use an external JNDI resource. An external JNDI factory must implement the `javax.naming.spi.InitialContextFactory` interface.

For information on creating an external JNDI resource, see Creating External JNDI Resources.

About Java Naming and Directory Interface (JNDI)

This section discusses the Java Naming and Directory Interface (JNDI), which is an application programming interface (API) for accessing different kinds of naming and directory services. J2EE components locate objects by invoking the JNDI lookup method.

This section covers the following topics:

- J2EE Naming Services
- Naming References and Binding Information
- Naming References in J2EE Standard Deployment Descriptor
- JNDI Connection Factories

J2EE Naming Services

A JNDI name is a user-friendly name for an object. These names are bound to their objects by the naming and directory service that is provided by a J2EE server. Because J2EE components access this service through the JNDI API, we usually refer to an object's user-friendly name as its JNDI name. For instance, the JNDI name of the Oracle database can be `jdbc/Oracle`. When it starts up, Sun ONE Web Server reads information from configuration file and automatically adds JNDI database names to the name space.

The application component's naming environment is a mechanism that allows customization of the application component's business logic during deployment or assembly. Use of the application component's environment allows the application component to be customized without the need to access or change the application component's source code.

A J2EE container implements the Web application component's environment, and provides it to the application component instance as a JNDI naming context. The application component's environment is used as follows:

- The Web application component's business methods access the environment using the JNDI interfaces. The application component provider declares in the deployment descriptor all the environment entries that the application component expects to be provided in its environment at runtime.
- The container provides an implementation of the JNDI naming context that stores the application component environment. The container also provides the tools that allow the deployer to create and manage the environment of each application component.
- A deployer uses the tools provided by the container to initialize the environment entries that are declared in the application component's deployment descriptor. The deployer can set and modify the values of the environment entries.
- The container makes the environment naming context available to the application component instances at runtime. The application component's instances use the JNDI interfaces to obtain the values of the environment entries.

Each application component defines its own set of environment entries. All instances of an application component within the same container share the same environment entries. Application component instances are not allowed to modify the environment at runtime.

Naming References and Binding Information

A resource reference is an element in a deployment descriptor that identifies the component's coded name for the resource. More specifically, the coded name references a connection factory for the resource. In the example given in the following section, the resource reference name is `jdbc/SavingsAccountDB`.

The JNDI name of a resource and the name of the resource reference are not the same. This approach to naming requires that you map the two names before deployment, but it also decouples components from resources. Because of this de-coupling, if at a later time the component needs to access a different resource, you don't have to change the name in the code. This flexibility also makes it easier for you to assemble J2EE applications from preexisting components.

The following table lists recommended JNDI lookups and their associated references for the J2EE resources used by Sun ONE Web Server.

Table 1 JNDI Lookups and Their Associated References

JNDI Lookup Name	Associated Reference
`java:comp/env`	Application environment entries
`java:comp/env/jdbc`	JDBC DataSource resource
`java:comp/env/mail`	JavaMail Session Connection Factories
`java:comp/env/url`	URL Connection Factories

Naming References in J2EE Standard Deployment Descriptor

A naming reference is a string used by the application to look up an object in the given naming context. For each Web application, there is a naming context and the references are configured in the standard component deployment descriptors.This section describes the standard deployment descriptor features used in Sun ONE Web Server. This section covers the following topics:

- Application Environment Entries
- References to Resources
- Resource Environment References

Application Environment Entries

Environment entries, defined using `<env-entry>`, provide a way of specifying deployment time parameters to J2EE Web applications. Note that the servlet context initialization parameters could be defined using `<context-param>`, but `<env-entry>` is the preferred way because application deployers to configure such applications parameters by explicitly specifying the name, type and values for them.

The following sample describes the syntax of <env-entry> as specified in the J2EE standard deployment descriptors:

```
<env-entry>
<description> Send pincode by mail </description>
<env-entry-name> mailPincode </env-entry-name>
<env-entry-value> false </env-entry-value>
<env-entry-type> java.lang.Boolean </env-entry-type>
</env-entry>
```

The <env-entry-type> tag specifies a fully qualified class name for the entry. Here is a code snippet to lookup the <env-entry> using JNDI from a servlet or JSP:

```
Context initContext = new InitialContext();
Boolean mailPincode = (Boolean)
initContext.lookup("java:comp/env/mailPincode");
// one could use relative names into the sub-context
Context envContext = initContext.lookup("java:comp/env");
Boolean mailPincode = (Boolean)
envContext.lookup("mailPincode");
```

References to Resources

A factory is an object that creates other objects on demand. A resource factory creates resource objects, such as database connections or message service connections. They are configured using <resource-ref> element in the standard deployment descriptors.

The following example describes the use of factories:

Example

Declaration of a reference to a JDBC connection factory that returns objects of type javax.sql.DataSource:

```
<resource-ref>
<description> Primary database </description>
<res-ref-name> jdbc/primaryDB </res-ref-name>
<res-type> javax.sql.DataSource </res-type>
<res-auth> Container </res-auth>
```

```
</resource-ref>
```

`<res-type>` is a fully-qualified class name of the resource factory. The `<res-auth>` variable can be assigned either `Container` or `Application` as a value.

If `Container` is specified, the web container handles the authentication before binding the resource factory to JNDI lookup registry. If `Application` is specified, the servlet must handle authentication programmatically. Different resource factories are looked up under a separate sub-context that describes the resource type, follows:

- `jdbc/` for a `JDBC javax.sql.DataSource` factory
- `mail/` for a JavaMail `javax.mail.Session` factory
- `url/` for a `java.net.URL` factory

Here is a code snippet to get JDBC connection from an application component with the container handling the authentication:

```
InitialContext initContext = new InitialContext();

DataSource source =

(DataSource) initContext.lookup("java:comp/env/jdbc/primaryDB");

Connection conn = source.getConnection();
```

Please note that in order to ensure that for these resource references work, the `res-ref-name` must map to valid resource factory at runtime.

Resource Environment References

Resource environment references provide a way of accessing, via JNDI lookups, administered objects associated with a resource. The `<resource-env-ref>` element, defined in the standard deployment descriptors lets applications declare the resource requirements.

The main difference between `<resource-env-ref>` and `<resource-ref>` element is the absence of specific resource authentication requirement; both these elements have to be backed up by a resource factory descriptor.

Examples

```
<resource-env-ref>

    <description> My Topic </description>

    <res-env-ref-name> jdbc/MyTopic </res-ref-name>

    <res-env-ref-type> javax.jdbc.Topic </res-type>
```

```
</resource-env-ref>
```

Here is a code snippet to access a JMS Topic object:

```
InitialContext initContext = new InitialContext();

javax.jms.Topic myTopic = (javax.jdbc.Topic)
initContext.lookup("java:comp/env/jdbc/MyTopic");
```

Initial Naming Context

The naming support in Sun ONE Web Server is based primarily on J2EE 1.3, with a few added enhancements.When an application component creates the initial context, via `InitialContext()`, Sun ONE Web Server returns an object that serves as a handle to the Web application's naming environment. This object in turn provides sub-contexts for the `java:comp/env` namespace. Each Web application gets its own namespace, that is, `java:comp/env` name space is per Web application and objects bound in one Web application's namespace don't collide with objects bound in other Web applications.

JNDI Connection Factories

For J2EE web applications, the deployment descriptor in the `web.xml` file is the placeholder for defining references to application environment entries or resource manager (such as SQL Data Source) connection factories. Applications look up such references using the `JNDI InitialNamingContext` provided by the J2EE containers. This makes applications portable to different Web Server environments by just making changes to the deployment descriptor, that is, without accessing or modifying the application's source code.

A connection factory is an object that produces connection objects that enable a J2EE component to access a resource. The connection factory for a database is a `javax.sql.DataSource object`, which creates a `java.sql.Connection` object.

In Sun ONE Web Server, you can configure the means of accessing the following resources and resource factories:

- JDBC connection factories
- JavaMail Session connection factories
- Generic, custom user-written resource object factories.
- Support for external resource repositories such as LDAP

All Sun ONE Web Server resource factories are specified within the `<resources> </resources>` tags in `server.xml` and have a JNDI name specified using the `jndiname` attribute (with the exception of `jdbconnectionpool` which does not have a jndiname). This attribute is used to register the factory in the server-wide namespace. Deployers can map user-specified, application-specific resource reference names (declared within `resource-ref` or `resource-env-ref` elements) to these server-wide resource factories using the `resource-ref` element in `sun-web.xml`. This enables deployment time decisions to be made with regards to which JDBC resources (and other resource factories) to use for a given application.

A custom resource accesses a local JNDI repository and an external resource accesses an external JNDI repository. Both types of resources need user-specified factory class elements, JNDI name attributes, and so on.

In this section, we will discuss how to create various J2EE resources, and how to access these resources.

Creating Java-based Resources

This section describes how you can use the Administration Interface to create different J2EE-based resources:

Creating a New JDBC Connection Pool

You can create a new JDBC connection pools in the following ways:

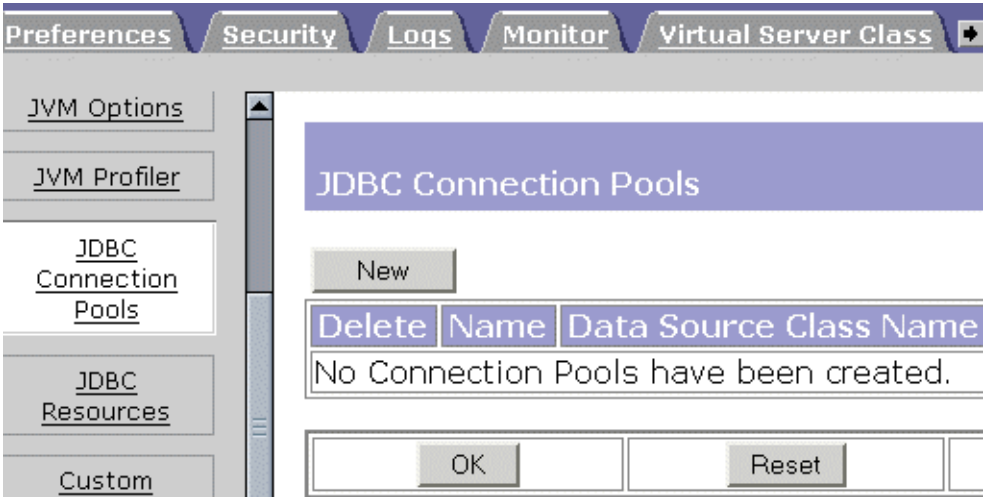
- Using the Administration Interface
- Using the Command-Line Interface

Using the Administration Interface

To create a new JDBC connection pool using the Administration interface, do the following:

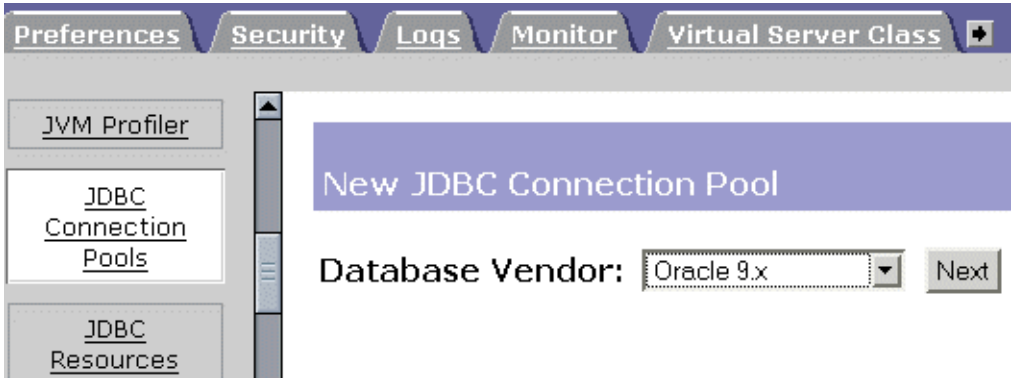
1. Access the Server Manager and choose the Java tab.
2. Click JDBC Connection Pools.
3. Click New.

   The JDBC Connection Pool Interface



4. From the Database Vendor drop down, select the type of database that you want to connect to. If your DBMS is not listed, select Other.

   The New JDBC Connection Pool Interface



5. Click Next.

   The Add New JDBC Connection Pool page is displayed.

6. Specify the properties for your new connection pool and click OK.

   Listed below are the connection pool properties that you must specify:

   General

   - **Pool Name**. Enter a name for the new connection pool.
   - **DataSource Classname**. The vendor-specific classname that implements the data source. If you selected Other from the Database Vendor list in the New JDBC Connection Pool page, you must enter the vendor-specific classname of the data source you plan to use. Please note that this class must implement `javax.sql.DataSource`.

   Properties

   Specify standard and proprietary JDBC connection pool properties; many of these properties are optional. By default the names of all of the standard properties are provided. You will need to consult your database vendor's documentation to determine which standard and vendor specific properties are required.

   Pool Settings

   - **Steady Pool Size**. Specify the minimum number of connections that the pool should maintain. When a connection is given to a requesting thread, it is removed from the pool, reducing the current pool size. The steady pool size also refers to the number of connections that will be added to the pool on server startup.
   - **Max Pool Size**. Specify the maximum number of connections that can be allowed in the pool at any given point in time.
   - **Pool Resize Quantity**. When the pool shrinks toward the steady pool size it is resized in batches. This value determines the size of the batch. Making this value too large will delay connection recycling, making it too small will be less efficient. Note, the pool capacity is only ever increased one connection at a time so this field does not effect increases in pool capacity.
   - **Idle Timeout (secs)**. The maximum time in seconds that a connection can remain idle in the pool. After this time, the pool implementation can close this connection.

- **Max Wait Time (milli secs)**. The amount of time the caller will wait before getting a connection timeout. The default wait time is `long`, which means that a caller can wait for a long time. If this value is set to `0`, the caller will be blocked until a connection is available.

Connection Validation

- **Connection Validation Required**. If this field is checked then connections will be validated before they are passed to the application. This allows the web server to automatically re-establish database connections in the case of the database becoming unavailable due to network failure or database server crash. Validation of connections will incur additional overhead and slightly reduce performance.
- **Validation Method**. Specifies the methods the Web server can employ to validate database connections. Choose from the following values:
  - **auto-commit**. In this mode, query statements are executed and committed as individual transactions. When `auto-commit` is disabled, query statements are grouped into transactions that can be terminated by either commit or roll back mechanisms.
  - **meta-data**. In this mode, a connection's database is able to provide meta-information describing its tables, its stored procedures, and so on. Each instance of the meta-data object will have a particular query associated with it. The meta-data object will execute that query and cache the results.
  - **table**. This method requires the Web server to perform a query on a user-specified table.
- **Table Name**. If you select the validation option, `table`, from the Validation Method drop-down list, specify the table name here.
- **Fail All Connections**. Specifies whether to fail all connections in the pool and re-establish them if a single connection is determined to have failed. If left unchecked, connections will be individually re-established only when they are used.

Transaction Isolation

The isolation level that a transaction uses determines how sensitive the application is to changes other users' transactions make, and consequently, how long the transaction must hold locks to protect against these changes.

- **Transaction Isolation**. Allows you to select the transaction isolation level for this connection. Choose from the following values:

- **read-uncommitted**. Also known as dirty read, this isolation level lets a transaction read any data currently on a data page, whether or not that data has been committed.
- **read-committed**. This places shared locks on data in such a way that data another transaction has changed but not yet committed will never be read. Because uncommitted data is not read, if a transaction running with `read-committed` isolation queries the data again, that data might have changed, or additional data might appear that meet the criteria of the original query.
- **repeatable-read**. This ensures that locks will be placed on all data that is used in a query. No other user can modify the data that your transaction visits as long as you have not yet committed or rolled back your transaction.
- **serializable**. This locks ranges of data so that if a query is reissued, no data will have changed and no additional rows of data will appear during the time interval between the first and second query.

- **Guarantee Isolation Level**. This ensures that any connection taken from the pool will have the same isolation level. For example, if the isolation level for the connection was changed programatically (for example, `con.setTransactionIsolation`) when last used, this mechanism will change it back to the specified isolation level.

Using the Command-Line Interface

For information on how to use the command-line interface to create a new JDBC connection pool, see Create JDBC Connection Pool in Appendix A, "Command Line Utilities".

Creating a JDBC Resource

A JDBC resource, also called a data source, lets you make connections to a database using `getConnection()`. Create a JDBC resource in one of these ways:

- Using the Administration Interface
- Using the Command Line Interface

Using the Administration Interface

To create a JDBC resource using the Administration interface, perform these tasks:

1. Access the Server Manager and choose the Java tab.

2. Click JDBC Resources.
3. Click the New button.
4. Enter the following information:
   - **JNDI Name** (required). Enter the JNDI name that application components must use to access the JDBC resource.
   - **Pool Name** (required). Select from the list the name (or ID) of the connection pool used by this JDBC resource. For more information, see Creating a New JDBC Connection Pool.
5. To enable the JDBC resource, select On from the Data Source Enabled drop-down.

   If a JDBC resource is disabled, no application component can connect to it, but its configuration remains in the server instance.
6. Click OK.
7. Click Apply Changes.

Using the Command Line Interface

For information on how to use the command-line interface to create a new JDBC resource, see Create JDBC Resource in Appendix A, "Command Line Utilities".

Creating Custom Resources

You can create a custom resource in any of the following ways:

- Using the Administration Interface
- Using the Command Line Interface

Using the Administration Interface

1. Access the Server Manager and choose the Java tab.
2. Click Custom Resources.
3. Click the New button.
4. Enter the following information:
   - **JNDI Name** (required). Enter the JNDI name that application components must use to access the custom resource.

- **Resource Type** (required). Enter the fully qualified type of the custom resource.
- **Factory Class** (required). Enter the fully qualified name of the user-written factory class, which implements `javax.naming.spi.ObjectFactory`.
- **Custom Resource Enabled** (optional). Select On to enable the custom resource at runtime.

5. Click OK.
6. Click Apply Changes.

Using the Command Line Interface

For information on how to use the command-line interface to create a new custom resource, see Create Custom Resource in Appendix A, "Command Line Utilities".

Creating External JNDI Resources

You can create an external resource in the following ways:

- Using the Administration Interface
- Using the Command Line Interface

Using the Administration Interface

1. Access the Server Manager and choose the Java tab.
2. Click External JNDI Resources.
3. Click the New button.
4. Enter the following information:
   - **JNDI Name** (required). Enter the JNDI name that application components must use to access the custom resource.
   - **Resource Type** (required). Enter the fully qualified type of the custom resource.
   - **Factory Class** (required). Enter the fully qualified name of the user-written factory class, which implements `javax.naming.spi.ObjectFactory`.

- **JNDI Lookup** (required). Enter the JNDI value to look up in the external repository. For example, if you are creating an external resource to connect to an external repository, to test a mail class, your JNDI Lookup could read `cn=testmail`.
- **External Resource Enabled** (optional). Select `On` to enable the external resource at runtime.

5. Click OK.
6. Click Apply Changes.

Using the Command Line Interface

For information on how to use the command-line interface to create a new custom resource, see Create External JNDI Resource in Appendix A, "Command Line Utilities".

Modifying Java-based Resources

This section describes how you can use the Administration interface to modify the properties of the Java-based resources you have created:

- Modifying a JDBC Connection Pool
- Modifying a JDBC Resource
- Modifying a Custom Resource
- Modifying an External JNDI Resource

Modifying a JDBC Connection Pool

To modify the properties of a JDBC connection pool:

1. Access the Server Manager and choose the Java tab.
2. Click the JDBC Connection Pools link.
3. Click the link representing the JDBC connection pool you want to edit.
4. Modify the settings as required.
5. Click OK.

Modifying a JDBC Resource

To modify the properties of a JDBC resource:

1. Access the Server Manager and choose the Java tab.
2. Click the JDBC Resources link.
3. Click the link representing the JDBC resource you want to edit.
4. Modify the settings as required.
5. Click OK.

Modifying a Custom Resource

To modify the properties of a custom resource:

1. Access the Server Manager and choose the Java tab.
2. Click the Custom Resources link.
3. Click the link representing the custom resource you want to edit.
4. Modify the settings as required.
5. Click OK.

Modifying an External JNDI Resource

To modify the properties of an external JNDI resource:

1. Access the Server Manager and choose the Java tab.
2. Click the External JNDI Resources link.
3. Click the link representing the external JNDI resource you want to edit.
4. Modify the settings as required.
5. Click OK.

Deleting Java-based Resources

This section describes how you can use the Administration interface to delete Java-based resources:

Deleting a JDBC Connection Pool

You can delete a JDBC reource using either of the following:

Using the Administration Server

To delete a JDBC connection pool using the Administration Server:

1. Access the Server Manager and choose the Java tab.
2. Click the JDBC Connection Pools link.
3. Check the checkbox corresponding to the JDBC connection pool you want to delete.
4. Click OK.

Using the Command Line Utility

For information on the syntax of the command line option you can use, see Command Line Utilities.

Deleting a JDBC Resource

You can delete a JDBC reource using either of the following:

Using the Administration Server

To delete a JDBC resource using the Administration server:

1. Access the Server Manager and choose the Java tab.
2. Click the JDBC Resources link.
3. Check the checkbox corresponding to the JDBC resource you want to delete.
4. Click OK.

Using the Command Line Utility

For information on the syntax of the command line option you can use, see Command Line Utilities.

Deleting a Custom Resource

You can delete a custom reource using either of the following:

- Using the Administration Server
- Using the Command Line Utility

Using the Administration Server

To delete a custom resource using the Administration server:

1. Access the Server Manager and choose the Java tab.
2. Click the Custom Resources link.
3. Check the checkbox corresponding to the custom resource you want to delete.
4. Click OK.

Using the Command Line Utility

For information on the syntax of the command line option you can use, see Command Line Utilities.

Deleting an External JNDI Resource

You can delete an external JNDI reource using either of the following:

- Using the Administration Server
- Using the Command Line Utility

Using the Administration Server

To delete an external JNDI resource using the Administration server:

1. Access the Server Manager and choose the Java tab.
2. Click the External JNDI Resources link.
3. Check the checkbox corresponding to the external JNDI resource you want to delete.
4. Click OK.

Using the Command Line Utility

For information on the syntax of the command line option you can use, see Command Line Utilities.

Part 4

Managing Virtual Servers and Services

Chapter 13

Using Virtual Servers

This chapter explains how to set up and administer virtual servers using your Sun ONE Web Server.

This chapter contains the following sections:

Virtual Servers Overview

When you use virtual servers you can offer companies or individuals domain names, IP addresses, and some server monitoring capabilities with a single installed server. For the users, it is almost as if they have their own web servers, though you provide the hardware and basic web server maintenance.

NOTE If you are not using virtual servers, you still use the items in the Class Manager to configure content, programs, and other features for your web server instance. When you install the web server, a default virtual server for the instance is created. You manage the content and services for this default virtual server using the virtual server user interface.

To set up virtual servers, you need to set up the following:

- Virtual Server Classes
- Listen Sockets
- Virtual Servers

The settings for virtual servers are stored in the `server.xml` file, found in the *server_root/server_ID*`/config` directory. You do not need to edit this file to use virtual servers, but you can. If you would like to learn more about this file and how to edit it, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference.*

This section includes the following topics:

- Multiple Server Instances
- Virtual Server Classes
- Listen Sockets
- Virtual Servers
- Virtual Server Selection for Request Processing
- Document Root
- Log Files
- Migrating Virtual Servers from a Previous Release

Multiple Server Instances

In past releases of the Sun ONE Web Server, unique configuration information for virtual servers was not very flexible. Quite often users created separate server instances in order to have a straightforward way to have servers with separate configuration information. The 6.0 version release of Sun ONE Web Server introduced separate configuration information for each virtual server class. Multiple server instances are still supported, but if your goal is to have many servers with separate configuration information, virtual servers are a better choice.

Virtual Server Classes

Virtual servers are grouped into classes. Using classes you can configure similar virtual servers at the same time, so you don't have to configure each one separately. Though all virtual servers in a class share the same basic configuration information, you can also set variables and change configuration per virtual server.

If you don't want virtual servers to share configuration information, you can create a single virtual server per virtual server class. However, if your virtual servers share similar properties, you can group them in a class and configure them together.

For example, if you work for an Internet Service Provider (ISP) and want to provide different levels of hosting for different customers at different prices, you can set up several classes of virtual servers for your customers. You might enable Java servlets and JSPs for one class of virtual servers, and disable Java servlets and JSPs for a less expensive class of virtual servers.

You create a class of virtual servers by naming it and setting up a document root, where all virtual servers belonging to the class will have their document roots by default. You can use the `$id` variable so that each virtual server within the class will have a separate document root within the class' document root. For more information, see "Document Root" on page 313.

After creating the class of virtual servers, you associate services with it. You can turn on or configure the following types of services for a class of virtual servers:

- Programs, see Extending Your Server With Programs.
- Content Management, see Content Management.
- Configuration Styes, see Applying Configuration Styles.

The obj.conf File

All virtual servers in a class share an `obj.conf` file, which stores information about the virtual server class. Some of that information is stored in variables, so that individual virtual servers can have specific variable values substituted on the fly.

For more information about `obj.conf` and variables, see the *NSAPI Programmer's Guide*. For more information on using variables in the user interface, see "Using Variables" on page 317.

Virtual Servers in a Class

A virtual server that belongs to a class is called a member of that class. Some virtual server settings are configured for all virtual servers in a class, and some are configured individually. These settings are configured on the Class Manager's Virtual Servers tab. For more information, see Chapter 14, "Creating and Configuring Virtual Servers".

The Default Class

When you install Sun ONE Web Server, the installer automatically creates a single class, called `defaultclass`. It contains one virtual server member by default for your server instance. You can add additional virtual servers to the default class, but you cannot delete your default virtual server from the class. You also cannot delete the default class.

Listen Sockets

Connections between the server and clients happen on a listen socket. Each listen socket you create has an IP address, a port number, a server name, and a default virtual server. If you want a listen socket to listen on all configured IP addresses on a given port for a machine, use 0.0.0.0, any, ANY, or INADDR_ANY for the IP address.

When you install Sun ONE Web 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 HTTP server port number during installation (the default is 80). You cannot delete the default listen socket. If you are not using virtual servers, this one listen socket is sufficient. However, if you are using virtual servers, you may want to create multiple listen sockets for your virtual servers.

Since a listen socket is a combination of IP address and port number, you can have multiple listen sockets with the same IP address and different port numbers, or with different IP addresses and the same port number. For example, you could have 1.1.1.1:81 and 1.1.1.1:82. Additionally, you could have 1.1.1.1:81 and 1.2.3.4:81, as long as your machine is configured to respond to both these addresses.

In addition, you specify the number of acceptor threads (sometimes called accept threads) in the listen socket. Acceptor threads are threads that wait for connections. The threads accept connections and put them in a queue where they are then picked up by worker threads. Ideally, you want to have enough accept threads so that there is always one available when a new request comes in, but few enough so that they do not provide too much of a burden on the system. The default is 1. A good rule is to have one accept thread per CPU on your system. You can adjust this value if you find performance suffering.

Virtual Servers

To create a virtual server you must first decide which class you want it to belong to. Next you need to decide what kind of virtual server you want. To create a virtual server, all you need to specify is a virtual server ID, and one or more URL hosts.

This section includes the following topics:

Types of Virtual Servers

Prior to the version 6.0 release of Sun ONE Web Server, there were two kinds of virtual servers: hardware and software. Hardware virtual servers had unique IP addresses associated with them. Software virtual servers did not have unique IP addresses but instead had unique URL hosts.

In Sun ONE Web Server 6.0 and Sun ONE Web Server 6.1, these concepts are no longer quite accurate. All virtual servers have a URL host specified. However, the virtual server may also be associated with an IP address based on its listen socket.

When a new request comes in, the server determines which virtual server to send it to based on the IP address or the value in the Host header. It evaluates the IP address first. For more information, see "Virtual Server Selection for Request Processing" on page 312.

IP-Address-Based Virtual Servers

In order to have multiple IP addresses on a single computer, you must either map them through the operating system or provide additional cards. To set up multiple IP addresses through the operating system, use the Network Control Panel (Windows) or the `ifconfig` utility (UNIX/Linux). Please note that directions for using `ifconfig` vary from platform to platform. Consult your operating system documentation for more information.

Typically you create an IP-address-based virtual server by creating a listen socket that listens on a specific IP address. The listen socket's default virtual server is an IP-address-based virtual server. For more information on ways to deploy virtual servers, see "Deploying Virtual Servers" on page 325.

URL-Host-Based Virtual Servers

You can set up URL-host-based virtual servers by giving them unique URL hosts. The contents of the Host request header directs the server to the correct virtual server.

For example, if you want to set up virtual servers for customers aaa, bbb, and ccc) so that each customer can have an individual domain name, you first configure DNS to recognize that each customer's URL, `www.aaa.com`, `www.bbb.com`, `www.ccc.com` , resolves to the IP address of the listen socket you are using. You then set the URL hosts for each virtual server to the correct setting (for example, `www.aaa.com`).

Because URL-Host-based virtual servers use the Host request header to direct the user to the correct page, not all client software works with them. Older client software that does not support the HTTP Host header won't work. These clients will receive the default virtual server for the listen socket.

Default Virtual Server

URL-Host-based virtual servers are selected using the Host request header. If the end user's browser does not send the Host header, or if the server cannot find the specified Host header, the default virtual server services the request.

The default virtual server is set by listen socket. You specify a default virtual server when you create a listen socket. You can always change the default virtual server.

Virtual Server Selection for Request Processing

Before the server can process a request, it must accept the request via a listen socket, then direct the request to the correct virtual server.

A virtual server is then selected as follows:

- If the listen socket is configured to only a default virtual server, that virtual server is selected.
- If the listen socket has more than one virtual server configured to it, the request `Host` header is matched to the URL host of a virtual server. If no `Host` header is present or no URL host matches, the default virtual server for the connection group is selected.

If a virtual server is configured to an SSL listen socket, its URL host is checked against the subject pattern of the certificate at server startup, and a warning is generated and written to the error log if they don't match.

After the virtual server is determined, the server executes the `obj.conf` file for the virtual server class to which the virtual server belongs. For details about how the server decides which directives to execute in `obj.conf`, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide.*

Document Root

The primary document directory or document root is the central directory that contains all the virtual server's files to make available to remote clients.

The document root directory provides an easy way to restrict access to the files on a virtual server. It also makes it easy to move documents to a new directory (perhaps on a different disk) without changing any of the URLs because the paths specified in the URLs are relative to the primary document directory.

For example, if your document directory is `C:\sun\servers\docs`, a request such as `http://www.sun.com/products/info.html` tells the server to look for the file in `C:\sun\servers\docs\products\info.html`. If you change the document root (that is, you move all the files and subdirectories), you only have to change the document root that the virtual server uses, instead of mapping all URLs to the new directory or somehow telling clients to look in the new directory.

When you install the Sun ONE Web Server, you designate a document root for your web server instance. That becomes the document root for the default class. You can change that directory at the class level or override it at the individual virtual server level.

When you add a class, you also need to specify a document directory. That directory is an absolute path. However, if you simply enter an absolute path, the document roots for all virtual servers belonging to the class default to the same directory. If you include the variable $id at the end of your document root absolute path, every virtual server has a default document root of *class_doc_root*/*virtual_server_ID*. For example, if your class' document directory is `/sun/servers/docs/$id`, the default document directory for a virtual server `vs1` that belongs to the class is `/sun/servers/docs/vs1`.

For more information on variables, see "Using Variables" on page 317.

You can also override the class' default document directory at the individual virtual server level.

Log Files

When you create a new virtual server, by default the log file is the same log file as the server instance. In most cases you will want each individual virtual server to have its own log file. To set this up, you can change the log path for each virtual server.

For more information, see "Configuring Virtual Server Log Settings" on page 338.

Migrating Virtual Servers from a Previous Release

If you used virtual servers in the 4.1 version of iPlanet Web Server, you can migrate them to the current release using the migration tool. For more information, see the *Installation and Migration Guide.*

Using Sun ONE Web Server Features with Virtual Servers

Sun ONE Web Server has many features, such as SSL and access control, that you can use with virtual servers. Many of these features involve configuration for all servers, for a server instance, for a class of virtual servers, or an individual virtual server. The following sections describe the features and provide information on where to look for more information.

This section includes the following topics:

Using SSL with Virtual Servers

If you want to use SSL on a virtual server, in most cases you use an IP-address-based virtual server. The customary port is 443. It is difficult to use SSL on a URL-host-based virtual server because Sun ONE Web Server must read the request before determining which URL host to send the request to. Once the server reads the request, the initial handshake, where security information is exchanged, has already happened.

The only exception is when URL-Host-based virtual servers all have the same SSL configuration, including the same server certificate, using “wildcard certificates.” For more information, see Chapter 6, “Using Certificates and Keys”.

One way to implement SSL with virtual servers is to have two listen sockets, one using SSL and listening to port 443, and one that is not using SSL. A user would typically access the virtual server through the non-SSL listen socket. When the need to have secure transactions arises, users could click a button on the web page to start initiating secure transactions. After that, the requests go through the secure listen socket.

Because SSL transactions are much slower than non-SSL transactions, this design limits the SSL transactions to only the ones that are necessary. Faster, non-SSL connections are used the rest of the time.

For more information on setting up and using security with your Sun ONE Web Server and virtual servers, see Chapter 6, "Using Certificates and Keys". For a diagram of a sample SSL configuration with virtual servers, see "Example 2: Secure Server" on page 327.

Using Access Control with Virtual Servers

With virtual servers you have the ability to set up access control on a per virtual server basis. You can even configure it so that each virtual server can have user and group authentication using an LDAP database. For more information, see "Controlling Access for Virtual Servers" on page 224.

Using CGIs with Virtual Servers

You can use CGIs on virtual servers. There are many settings that you can configure on for access and security reasons.

For more information on setting up and using CGIs, see "Installing CGI Programs" on page 355.

Using Configuration Styles with Virtual Servers

Configuration styles are an easy way to apply a set of options to specific files or directories that your various virtual servers maintain. For more information on using configuration styles see Applying Configuration Styles.

Using the Virtual Server User Interface

To create and edit virtual servers, you can use the user interface or a command line utility.

The user interface for administering virtual servers has three parts:

- The Server Manager contains settings that affect the server as a whole (or all virtual servers).
- The Class Manager contains settings that affect a single class and the virtual servers within the class.
- The Virtual Server Manager contains settings for an individual virtual server.

In addition, a user interface for end-users who have an individual virtual server is available. For more information, see "Allowing Users to Monitor Individual Virtual Servers" on page 321.

This section includes the following topics:

The Class Manager

To access the Class Manager follow these steps:

1. From the Server Manager, click the Virtual Server Class Tab.
2. Click Manage Classes.
3. Choose a class and click Manage.

You can also click the class name in the tree view of the server, or click the Class Manager button link in the upper right corner of the Server Manager.

The Virtual Server Manager

To access the Virtual Server Manager, follow these steps:

1. From the Class Manager, click the Virtual Server Tab.

2. Click Manage Virtual Servers.
3. Choose a virtual server and click Manage.

You can also click the virtual server name in the tree view of the server.

You can use a command line utility, `HttpServerAdmin`, to perform the same virtual server tasks as you can perform using the user interface. For more information on the command line utility `HttpServerAdmin`, see "HttpServerAdmin (Virtual Server Administration)" on page 441.

Using Variables

You can use variables to give virtual-server specific values for a class without having to define each value individually. A variable is defined in the `obj.conf` file. You can define your own variables, but the user interface will not recognize them. The variable that is most useful in the user interface is the variable $id, which represents the ID of the virtual server. Whenever you enter this variable, the server substitutes the value for the individual virtual server ID.

There are a few other variables, such as $accesslog (the path to each virtual server's access log) and $docroot (the path to each virtual server's document root), that you may occasionally see, but $id is the only one you should need to enter into a field.

For more information on variables, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide.*

Dynamic Reconfiguration

Dynamic reconfiguration allows you to make configuration changes to a live web server without having to stop and restart the web server for the changes to take effect. You can dynamically change all configuration settings and attributes in `server.xml` and its associated files without restarting the server. So any changes that you make within the virtual server user interface can by applied without restarting the server. You can dynamically reconfigure your server after changes using the reconfiguration script or the user interface.

On UNIX platforms the dynamic reconfiguration script is a shell script named '`reconfig`' located in each instance's directory. There are no commandline arguments to this script. You can run the reconfiguration script by simply typing '`reconfig`' from the server instance's directory.

On Windows, the dynamic reconfiguration script is a batch file called 'reconfig.bat located in each instance's directory. There are no command line arguments. You can run the reconfiguration script by simply typing 'reconfig' or 'reconfig.bat' from the server instance's directory.

When run, this script initiates a dynamic reconfiguration of the server, similar to the user interface, and displays the server messages related to reconfiguration.

To access the dynamic reconfiguration screen, click the Apply link found in the upper right corner of the Server Manager, Class Manager, and Virtual Server Manager pages, then click the Load Configuration Files button on the Apply Changes page. If there are errors in installing the new configuration, the previous configuration is restored.

Setting Up Virtual Servers

To set up virtual servers, follow these steps:

1. Create a listen socket
2. Create a class of virtual servers
3. Configure the services for the class
4. Create the virtual servers in a virtual server class
5. Configure virtual servers

Please note that you must enter an existing virtual server in the default virtual server field when you create a listen socket. You can use the virtual server created when you installed the server, and then go back and change it after you've created additional virtual servers, if you like.

Creating a Listen Socket

To create a listen socket, follow these steps:

1. From the Server Manager, click the Preferences tab.
2. Click Add Listen Socket.

3. Fill in the fields.

   Listen sockets must have a unique combination of port number and IP address. You can use either IPV4 or IPV6 addresses. If you want to create a listen socket for IP-address-based virtual servers, the IP address must be 0.0.0.0, ANY, any or INADDR_ANY, meaning it listens on all IP addresses on that port.

   You can also enable security (SSL) for this listen socket.

   The Server Name field specifies the host name in the URLs the server sends to the client. This affects URLs the server automatically generates; it doesn't affect the URLs for directories and files stored in the server. This name should be the alias name if your server uses an alias.

4. Click OK.

Creating a Virtual Server Class

To create a virtual server class, follow these steps:

1. From the Server Manager, click the Virtual Server Class tab.
2. Click Add Class.
3. Name the class.
4. Insert a document root for the class.

   The directory must already exist. All virtual servers for this class will have document roots in this absolute path, unless you specify otherwise. If you use `/$id` as the last part of the path, a document root folder named for the virtual server ID is automatically created within the class' document root path.

5. Click OK.

Once you have created a class of virtual servers, choose the services associated with the class. For more information, see Content Management.

Editing or Deleting a Virtual Server Class

To edit a virtual server class's settings, follow these steps:

1. From the Server Manager, click the Virtual Server Class tab.
2. Click Edit Classes.

3. From the drop-down list next to the class you want, choose Edit or Delete.

   Please note that you cannot delete the default class.

4. Use the Document Root field to change to absolute path to the class' default document root.

   The document roots for virtual servers in this class are created within this directory by default.

5. Enter On in the Accept Language field if you wish this class of virtual servers to use accept language header parsing.

   The default is Off.

6. If you want to change the CGI defaults associated with a class, click Advanced.

   A window with the CGI defaults appears. Edit the fields and click OK to return to the Edit a Class window. The Reset button rolls back your changes.

7. Click OK. The class is changed or deleted.

Specifying Services Associated with a Virtual Server Class

Some of the characteristics that differentiate one class of virtual servers from another are the services that are enabled for that class of virtual servers. For example, one class of virtual servers might have CGIs enabled while another doesn't. For more information on setting up services, see Content Management.

Creating a Virtual Server

Once you have set up a virtual server class, you can create a virtual server. Because virtual servers are members of a particular virtual server class, you create virtual servers on the Class Manager.

For more information, see "Creating a Virtual Server" on page 333.

Specifying Settings Associated with a Virtual Server

You can override some class settings at the virtual server level and also configure additional settings. You configure these settings in the Class Manager.

For more information, see “Creating a Virtual Server” on page 333.

Allowing Users to Monitor Individual Virtual Servers

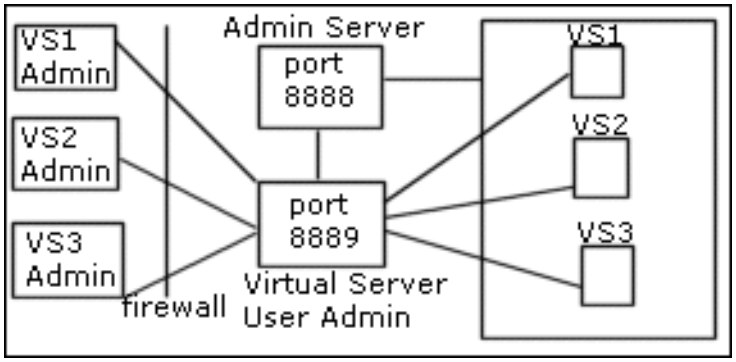
A special user interface exists for the administrators of individual virtual servers that allows them to see settings for their virtual servers and to view their access and error logs. For example, if you have an intranet with three different virtual servers for three different departments, each department can view their settings and log files individually.

For security reasons, this administration user interface is on a separate port from either the administration server port or the web server instance port.

This user interface runs on a virtual server within the administration server. This virtual server is set up by default and is called useradmin. You must set up a listen socket in the administration server that is separate from the listen socket the administration server runs on, so that people can access the virtual server administration user interface without having access to your administration server port.

The following figure, , shows the administrators of individual virtual servers accessing the useradmin virtual server in order to access the information for their virtual servers.

Configuring virtual server administrator's user interface



When you turn on a virtual server, if you edit certain settings in the Administration Server's `/config/server.xml` file, users can administer it, through the following URL:

server_name`:`*port*`/user-app/`*server_instance*`/`*virtual_server_ID*

For example:

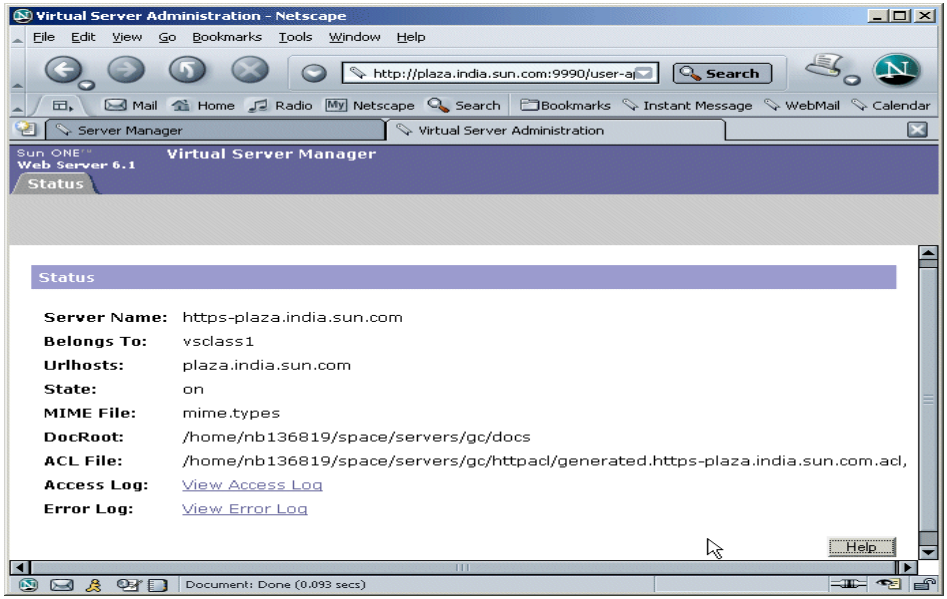
`sun:9999/user-app/sun/vs2`

The server instance doesn't include the "`https`" portion of the server instance name.

To determine the virtual server ID, look up the `server.xml` file of the server instance.

The following figure shows the user interface that the end users see:

Virtual Server Administration user interface



After you install Sun ONE Web Server 6.1, you will find that the *server_root*`/https-admserv/config/server.xml` file contains certain commented-out entries that create:

- a default listen socket for a virtual server called useradmin.
- a virtual server class for the virtual server.

To set up useradmin, all you need to do is to uncomment these entries.

To configure your server to use this feature, follow these steps:

1. Create a new listen socket that runs a port separate from the port that the administration server uses.

   For example, if your administration server runs on port 8888, this new listen socket must have a different port number. Using a different listen socket helps safeguard your administration server.

   For security reasons, you cannot add this listen socket through the user interface. Instead, you add it in the administration server's `server.xml` file.

2. Open the administration server's `server.xml` file, found at *server_root*`/https-admserv/config/server.xml`.

3. Uncomment the commented lines containing default values for the LS, VSCLASS, and VS elements. Example:

```
<!--
<LS id="ls2" port="9999" servername="plaza"
defaultvs="useradmin"/>
-->
<!--
<VSCLASS id="userclass" objectfile="userclass.obj.conf">
   <VS id="useradmin" connections="ls2" mime="mime1"
aclids="acl1" urlhosts="plaza">
      <PROPERTY name="docroot" value="/export1/wsinst/docs"/>
      <USERDB id="default"/>
      <WEBAPP uri="/user-app"
path="/export1/wsinst/bin/https/webapps/user-app"/>
   </VS>
</VSCLASS>
-->
```

   This will enable useradmin, created on a separate port for security reasons.

4. Save your changes to `server.xml`.
5. Apply the changes by restarting the Administration Server.
6. For any virtual server in any server instance, you should now be able to access the administrator UI by using the following URL:

   server_name:*port*/user-app/*server_instance*/*virtual_server_ID*

   For example:

   `plaza:9999/user-app/plaza/https-plaza`

Access Control

To protect the virtual server administration from unauthorized users, you can set up ACLs. Because the URI for each virtual server is unique, you can set access so that only the correct administrator can access the settings for a virtual server.

For more information, see Chapter 9, “Controlling Access to Your Server”.”

Log Files

Each virtual server can have its own log files. By default, all virtual servers share the log file of the server instance. If you allow users to view their log files, in most cases you should change the log file settings so that each virtual server has its own access and error log.

For more information, see “Configuring Virtual Server Log Settings” on page 338.

Deploying Virtual Servers

Sun ONE Web Server’s virtual server architecture is very flexible. A server instance can have any number of listen sockets, both secure and non-secure. You can have both IP-address-based and URL-host-based virtual servers.

In addition, you can group virtual servers with similar settings into any number of virtual server classes. All virtual servers in a virtual server class share the same request processing instructions in `obj.conf`.

Every virtual server can (but does not have to) have its own list of ACLs, its own `mime.types` file, and its own set of Java Web Applications.

This design gives you maximum flexibility to configure the server for a variety of applications. The following examples discuss some of the possible configurations available for Sun ONE Web Server.

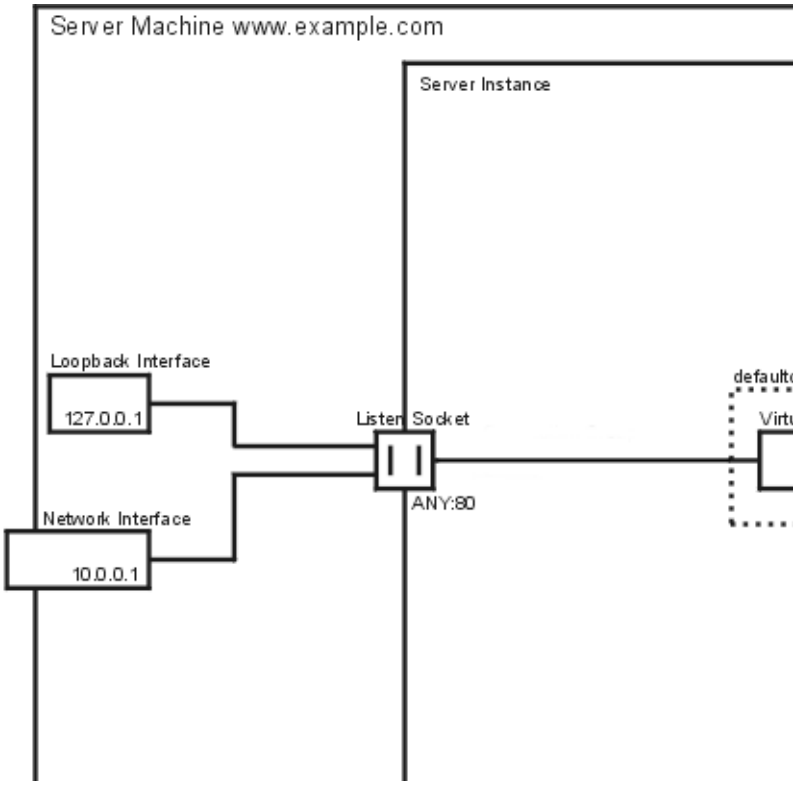
Example 1: Default Configuration

After a new installation of the Sun ONE Web Server, you have one server instance. This server instance has just one listen socket listening on port 80 (or whatever you selected at installation) of any IP address to which your computer is configured.

Some mechanism in your local network establishes a name-to-address mapping for each of the addresses to which your computer is configured. In the following example, the computer has two network interfaces: the loopback interface (the interface that exists even without a network card) on address 127.0.0.1, and an ethernet interface on address 10.0.0.1.

The name example.com is mapped to 10.0.0.1 via DNS. The listen socket is configured to listen on port 80 on any address to which that machine is configured ("ANY:80" or "0.0.0.0:80").

Default configuration



In this configuration, connections to the following reach the server and are served by virtual server VS1

- `http://127.0.0.1/` (initiated on `example.com`)
- `http://localhost/` (initiated on `example.com`)
- `http://example.com/`
- `http://10.0.0.1/`

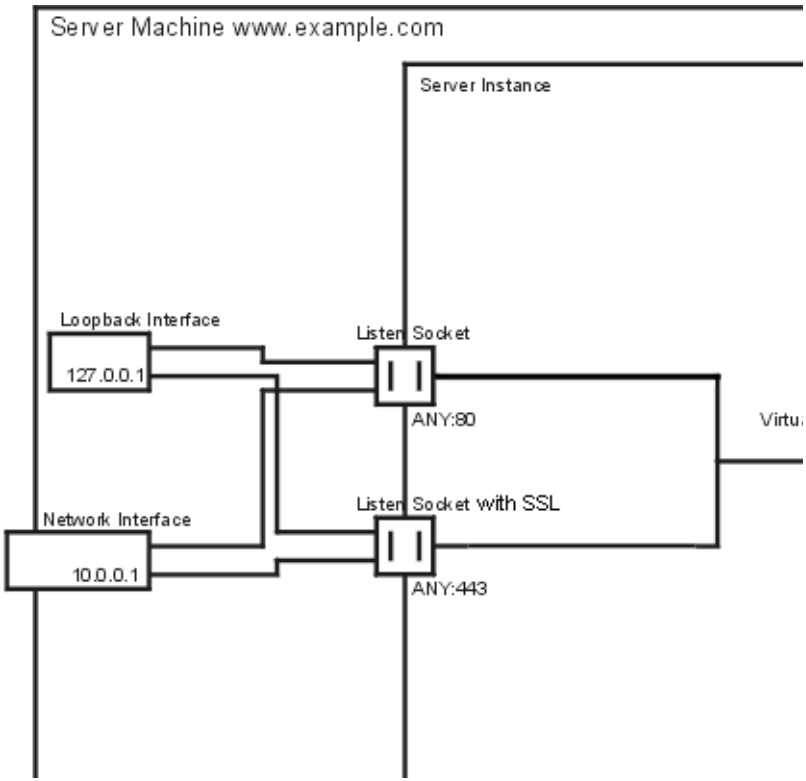
Use this configuration for traditional web server use. You do not need to add additional virtual servers or listen sockets. You configure the settings of the server by changing the settings for defaultclass (VS1 is a member of defaultclass), and VS1 itself.

Example 2: Secure Server

If you want to use SSL in the default configuration, you can simply change the listen socket to secure mode. This is a similar to the way you set security in previous versions of the Sun ONE Web Server.

You can also add a new secure listen socket configured to ANY:443 and associate VS1 to the new listen socket. The virtual server now has listen sockets, one that uses SSL, and one that doesn't. Now your server will serve the same content both with and without SSL, i.e. `http://example.com/` and `https://example.com/` deliver the same content.

Secure server



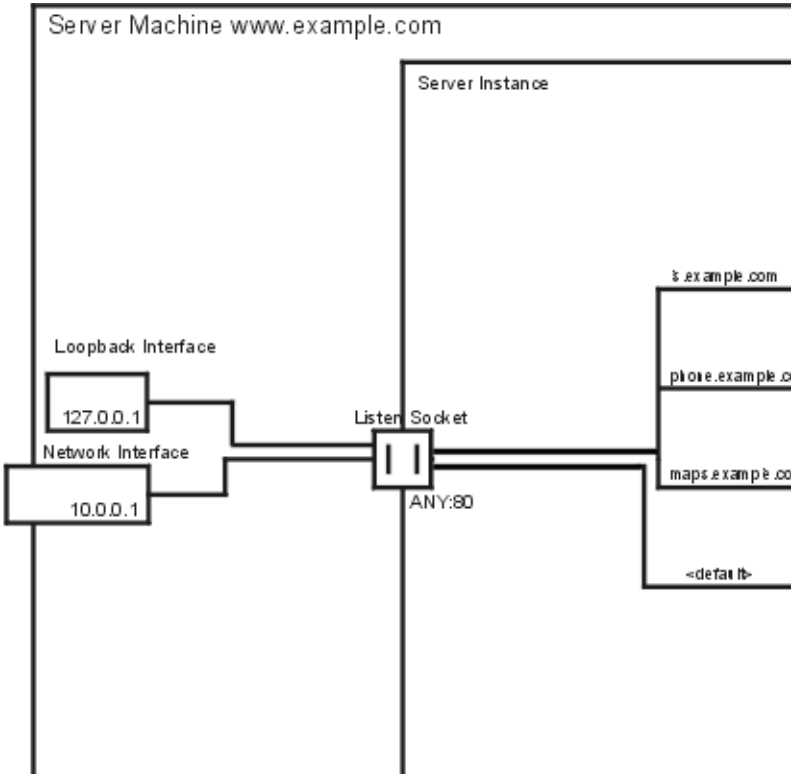
Please note that the SSL parameters are attached to the listen socket. Therefore, there can only be one set of SSL parameters for all the virtual servers configured to a particular listen socket.

Example 3: Intranet Hosting

A more complex configuration of the Sun ONE Web Server is one in which the server hosts a few virtual servers for an intranet deployment. For example, you have three internal sites where employees can look up other users' phone numbers, look at maps of the campus, and track the status of their requests to the Information Services department. Previously (in this example), these sites were hosted on three different computers that had the names `phone.example.com`, `maps.example.com` and `is.example.com` mapped to them.

To minimize hardware and administrative overhead, you want to consolidate all three sites into one web server living on the machine `example.com`. You could set this up in two ways: using URL-host-based virtual servers or using separate listen sockets. Both have their distinct advantages and disadvantages.

Intranet hosting using URL-host-based virtual servers

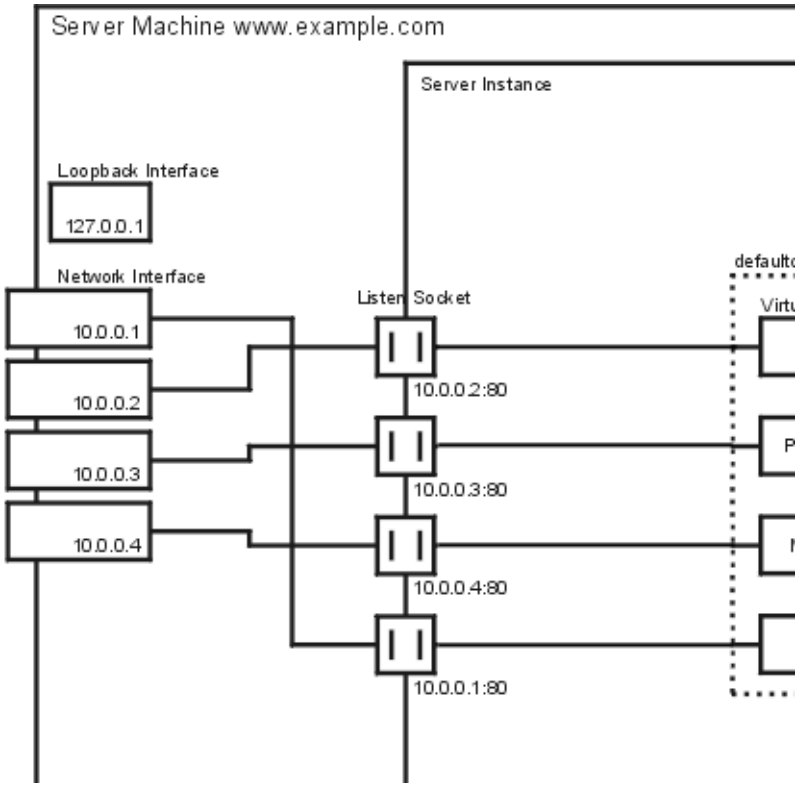


While URL-host-based virtual servers are easy to set up, they have the following disadvantages:

- Supporting SSL in this configuration requires non-standard setup using wildcard certificates. For more information see Chapter 4, “J2EE-based Security for Web Container and Web Applications”.
- URL-host-based virtual servers don’t work with legacy HTTP clients

You could also set up the IP-address-based configuration with one listen socket per address:

Intranet hosting using separate listen sockets



The advantages to IP-address-based virtual servers are:

- They work with older clients that do not support the HTTP/1.1 Host header.
- Providing SSL support is straightforward.

The disadvantages are:

- They require configuration changes on the host computer (configuration of real or virtual network interfaces)

- They don’t scale to configurations with thousands of virtual servers

Both configurations require setting up name-to-address mappings for the three names. In the IP-address-based configuration, each name maps to a different address. The host machine must be set up to receive connections on all these addresses. In the URL-host-based configuration, all names can map to the same address, the one the machine had originally.

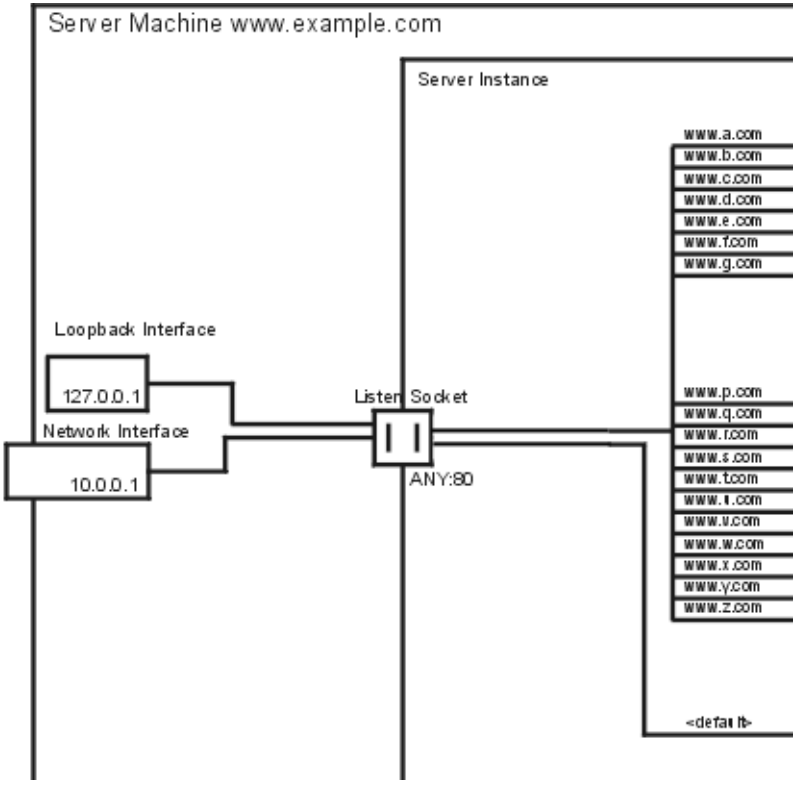
The configuration with multiple listen sockets may give you a minimal performance gain because the server does not have to find out the address the request came in on. However, using multiple listen sockets also results in additional overhead (memory and scheduling) because of the additional acceptor threads.

Example 4: Mass Hosting

Mass hosting is a configuration in which you enable many low-traffic virtual servers. For example, an ISP that hosts many low-traffic personal home pages would fall into this category.

The virtual servers are usually URL-host-based and are in one of multiple virtual server classes, depending on the level of service provided. For example, you could have one class that allows only static content, and another one that allows static content plus CGIs.

Mass Hosting



Notice that the virtual server installed when you installed the server, VS1, still exists in `defaultclass`.

Chapter 14

Creating and Configuring Virtual Servers

A class of virtual servers has virtual servers (members of the class) associated with it. You can override some of the class-level settings at the virtual server level. This chapter describes how you can create and configure individual virtual servers. For information on configuring virtual server classes, see Content Management. For an overview of virtual servers, see Using Virtual Servers.

This chapter contains the following sections:

Creating a Virtual Server

Virtual servers allow you, with a single installed server, to offer companies or individuals domain names, IP addresses, and some server administration capabilities. For an introduction to virtual servers and how to set them up in the Sun ONE Web Server, see Using Virtual Servers.

To create a virtual server, follow these steps:

1. From the Class Manager, choose the Virtual Servers tab.
2. Click Add Virtual Server.
3. Choose a name for the virtual server.

4. Choose a URL host for the virtual server.

   You can type more than one URL host, separated by spaces.

5. Click OK.

These settings are all that is required for creating a virtual server. However, you can configure additional virtual server settings using other pages on this tab.

Editing Virtual Server Settings

Once you have set up your virtual servers, you can edit them. You can make these changes two ways: using the Class Manager or the Virtual Server Manager.

On the Class Manager, the pages are organized by the kind of setting you want to change. For example, you can go to the Quality of Service page to change the Quality of Service settings for one or more virtual servers in the class.

On the Virtual Server Manager, the pages only pertain to one virtual server, so you can see and change all of its settings.

Editing Using the Class Manager

Use the following Class Manager pages to edit virtual server settings.

Editing Virtual Server Settings

To edit the general settings of a virtual server, use the Edit Virtual Servers page. To access this page, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.
2. Click Edit Virtual Servers.
3. To edit a virtual server, click From the drop-down list next to the virtual server you want, choose Edit or Delete.

   The default virtual server can only be edited and not deleted.

4. Set the State to On, Off, or Disabled.

   If you set the state to Disabled, you can turn the server back on, but the end user of the server cannot.

   This state is the virtual server's state, which is independent of whether the server instance is on or off. If a virtual server's state displayed on this page is on, the virtual server can only accept requests if the server instance is on as well.

   This is true of the default virtual server for the default server instance as well. If you turn off your server instance, your default virtual server is still set to on, but will not accept connections.

   You cannot turn off or disable the default virtual server for the server instance.

5. Type the URL Hosts you want to use, if different than displayed under Urlhosts column.

   You can type more than one URL host, separated by spaces.

6. When you are through editing virtual servers click OK.

Configuring Virtual Server MIME Settings

You can set the MIME types file for an individual virtual server. The MIME types file contains the mappings of file extensions to types of files. For example, the MIME types file is where you can specify that all files ending `.cgi` be treated as CGI files.

You don't need to create a separate MIME types file for each virtual server or virtual server class. Instead, you create as many MIME types files as you need and associate them with a virtual server. One MIME types file, `mime.types`, exists by default on the server. To create new MIME types files, or to edit the definitions in a MIME Types file, see "Choosing MIME Types" on page 175.

To set the MIME types file for a specific virtual server, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.
2. Click MIME Settings.
3. Choose a MIME types file from the drop-down list next to the virtual server.
4. Click OK.

Configuring Virtual Server ACL Settings

You can use ACLs to control access to virtual servers. Each virtual server can have a different base DN in the LDAP database, so that each virtual server can have its own entries in a the single LDAP database used by the Sun ONE Web Server.

For more information, see "Controlling Access for Virtual Servers" on page 224.

Configuring Virtual Server Security

You can set security for a virtual server if that virtual server is bound to a secure listen socket.

For more information on security, see Chapter 4, "J2EE-based Security for Web Container and Web Applications".

Configuring Virtual Server Quality of Service Settings

Quality of service refers to the performance limits you set for a virtual server. For example, an ISP might want to charge different amounts of money for virtual servers depending on how much bandwidth allowed them.

You can enable these settings for the entire server or for a class of virtual servers in the Server Manager, from the Status tab. However, you can override these server or class-level settings for an individual virtual server.

Before enabling quality of service for a virtual server, you must first enable it for the entire server, and also set some basic values. See "Using Quality of Service" on page 253.

To configure the quality of service settings for a virtual server, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.
2. Click Quality of Service.

   A page appears listing all the virtual servers in the class and their quality of service settings.

3. To enable quality of service for a virtual server, choose Enable from the drop-down list.

   By default quality of service is disabled. Enabling quality of service increases server overhead slightly.

4. Set the maximum bandwidth, in bytes per second, for the virtual server.

5. Choose whether or not to enforce the maximum bandwidth setting.

   If you choose to enforce the maximum bandwidth, once the server reaches its bandwidth limit additional connections are refused.

   If you do not enforce the maximum bandwidth, when the maximum is exceeded the server logs a message to the error log.

6. Choose the maximum number of connections allowed for the virtual server.

   This number is the number of concurrent requests processed.

7. Choose whether or not to enforce the maximum connections setting.

   If you choose to enforce the maximum connections, once the server reaches its limit additional connections are refused.

   If you do not enforce the maximum connections, when the maximum is exceeded the server logs a message to the error log.

8. Click OK.

For more information on the limitations to the quality of service features, see "Using Quality of Service" on page 253.

Configuring Virtual Server Log Settings

To change the location of the virtual server's access and error logs from the default, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.
2. Click Logging Settings.

   A page appears listing all the virtual servers in the class and the location of their error logs.

3. Enter an absolute path to the error and access logs. The path must already exist.

   By default, the access and error messages for all virtual servers are logged to the server instance's access and error logs. If you want virtual servers to have separate log files, you set that up here.

4. If you want to change the paths back to the default, click Default.
5. Click OK.

To look at the logs for a particular virtual server, follow these steps:

1. From the Virtual Server Manager choose the Logs tab.
2. Click View Access Log or View Error Log.
3. Choose the number of entries to display and the criteria for displaying them.

   For example, if your logs contain entries for all virtual servers, you can choose to display only the entries for a particular virtual server.
4. Click OK.

Enabling Logging for a Virtual Server

To enable virtual server level logging, perform the following steps:

1. Go to the Logs tab in the Server Manager for the server instance and select Log Preferences.
2. Create a new access log by entering the path and file name in the Log File field.

   You can also manually create a new access log in `magnus.conf` by changing

   `Init fn=init access="$accesslog"` to `Init fn=init access="newaccesslog"`
3. Select Only Log under Format and check Virtual Server Id.

   For a custom format, select Custom Format and add `%vsid%` to the end of the line.

   `%vsid%` is useful when using multiple virtual servers. This entry records vsid in the access log.

   You can also manually add `%vsid%` to the end of `Init fn` in the `magnus.conf` file.
4. Click OK.
5. Click Apply.
6. Click Apply Changes for your changes to take effect.

Configuring Virtual Server Java Web Application Settings

A web application is a collection of Java servlets, JSPs, HTML pages, classes and other resources. All the resources are stored in a directory, and all requests to that directory run the application. Use the pages under the Web Applications tab of the Virtual Server Manager to deploy and edit web applications for a specific virtual server.

For more information on web applications and the deployment descriptor file for web applications,`sun-web.xml`, see the Sun ONE Web Server 6.1 *Administrator's Configuration File Reference*.

Editing Using the Virtual Server Manager

The Virtual Server Manager contains four tabs: Preferences, Logs, Web Applications, and WebDAV.

The Preferences tab contains pages for:

- Status
- Settings

The Status page lists some settings and provides links to the virtual server's access and error logs.

The Settings page contains the following settings for a virtual server:

- State (on or off)
- Document root
- Access and error log directories
- Directory services
- ACL file
- MIME types file
- CGI settings

If you are editing a single virtual server, it's convenient to use the Virtual Server Manager and change all these settings on one page.

The Logs tab contains a single page allowing you to generate reports for the selected virtual server.

For more information about deploying and editing web application files, see Chapter 15, “Extending Your Server With Programs” .

The WebDAV Tab allows you to create and edit WebDAV collections on a virtual server. A WebDAV collection is a resource or a set of resources that are enabled for WebDAV operations. Using WebDAV, you can collaboratively author documents in-place on the Web. WebDAV allows you to place locks of different levels of granularity on WebDAV-enabled resources thus effectively preventing overwrite conflicts during collaborative content authoring on the Web.

The WebDAV tab contains the following pages:

- The Add Collection Page
- The Edit DAV Collection Page
- The Lock Management Page

The Add Collection page allows you to create a WebDAV collection.

The Edit DAV Collection page allows you to configure WebDAV-enabled collections.

The Lock Management page allows you to view outstanding locks and other lock-related information pertaining to the WebDAV-enabled resources on your server.

For more information, see Chapter 19, “Web Publishing with WebDAV”.

Generating Reports for a Virtual Server

You can now generate a report for a single virtual server using the Virtual Server Manager. To do so, you should first create a new access log to be used by the virtual server, and add the new access log to the virtual server settings, as described below.

To generate a report for a virtual server, follow these steps:

1. Go to the Logs tab in the Server Manager for the server instance and select Log Preferences.

2. Create a new access log by entering the path and file name in the Log File field.

   You can also manually create a new access log in magnus.conf by changing

   `Init fn=init access="$accesslog"` to `Init fn=init access="newaccesslog"`

3. Select Only Log under Format and check Virtual Server Id.

   For a custom format, select Custom Format and add `%vsid%` to the end of the line.

   `%vsid%` is useful when using multiple virtual servers. This entry records vsid in the access log.

   You can also manually add `%vsid%` to the end of `Init fn` in the `magnus.conf` file.

4. Click OK.
5. Click Apply.
6. Click Apply Changes for your changes to take effect.
7. Select the virtual server you wish to generate a report for and go to the Virtual Server Manager > Manage Classes > select the Virtual server from the tree view.
8. Go to the Preferences tab and select Settings.

   In the Access Log field, change the access log to the newly created one.

9. Click OK.
10. Click Apply.
11. Click Apply Changes for your changes to take effect.
12. Select the Logs tab.

    The Generate Reports page appears.

    This page will not appear unless a virtual server has been created and `LogVSid` is On. For more information about enabling the Virtual Server Id, see Enabling Logging for a Virtual Server.

13. (Optional) change the settings if desired.
14. Click OK to generate the report.

Choosing a Directory Service for a Virtual Server

You can assign a particular directory service for a particular virtual server. When you do so, the directory service you choose is logged under the `USERDB` element of the corresponding `VS` (virtual server) element in the `server.xml` file. The rights and permissions associated with this directory service is later used by the server to evaluate and enforce access control rules.

To assign a directory service to a virtual server, perform the following steps:

1. From the Virtual Server Manager choose the Settings tab.

   A listing of the virtual server settings is displayed.

2. Click the Edit link next to Directory Services.

   The Pick Directory Services for Virtual Server page is launched in a new window.

3. Choose a directory service and click OK.

4. Save and apply changes.

NOTE The directory service you choose for a particular virtual server is not shared across other virtual servers. Access control files, on the other hand, are shared across virtual servers.

Deleting a Virtual Server

To delete a virtual server, follow these steps:

1. From the Class Manager, click the Virtual Servers tab.

2. Click Edit Virtual Servers.

3. From the drop-down list next to the virtual server you want, choose Delete.

   You cannot delete the default virtual server that was created when you installed the server.

4. Click OK.

   The virtual server is deleted.

Chapter 15

Extending Your Server With Programs

This chapter discusses how to install programs on the Sun ONE Web Server that dynamically generate HTML pages in response to requests from clients. These programs are known as *server-side applications*. (*Client-side applications* are downloaded to the client and run on the client machine.)

This chapter includes the following sections:

Overview of Server-Side Programs

Java servlets and CGI programs have different strengths and uses. The following list illustrates the differences between these server-side programs:

- Java servlets are written in Java, which is a full-featured programming language for creating network applications.
- CGI (Common Gateway Interface) programs can be written in C, Perl, or other programming languages. All CGI programs have a standard way of passing information between clients and servers.

Types of Server-Side Applications That Run on the Server

The Sun ONE Web Server can run the following types of server-side applications to dynamically generate content:

- Java servlets
- CGI programs

The Sun ONE Web Server can also run programs that extend or modify the behavior of the server itself. These programs, known as plug-ins, are written using the Netscape Server Application Programming Interface (NSAPI). For information about writing and installing plug-in programs, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide.*

How Server-Side Applications Are Installed on the Server

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

- For Java servlets, you can create and deploy web applications. For more information, see "What the Server Needs to Run Servlets" on page 348.
- For CGI programs, you can configure your server to recognize all files with certain filename extensions, or all files in specified directories as CGI programs, or both. For more information, see "Installing CGI Programs" on page 355, "Installing Windows CGI Programs" on page 359, and "Installing Shell CGI Programs for Windows" on page 362.

These installation procedures are described in the following sections.

Java Servlets and JavaServer Pages (JSP)

This section discusses how to install and use Java Servlets and JavaServer Pages on Sun ONE Web Server.

The following topics are described:

Overview of Servlets and JavaServer Pages

Sun ONE Web Server 6.1 supports the Servlet 2.3 API specification, which allows servlets and JSPs to be included in web applications.

A web application is a collection of servlets, JavaServer Pages, HTML documents, and other web resources which might include image files, compressed archives, and other data. A web application may be packaged into an archive (a WAR file) or exist in an open directory structure.

NOTE Servlet API version 2.3 is fully backward compatible with version 2.1, so all existing servlets will continue to work without modification or recompilation.

To develop servlets, use Sun Microsystems' Java Servlet API. For information about using the Java Servlet API, see the documentation provided by Sun Microsystems at:

`http://java.sun.com/products/servlet/index.html`

A JSP is a page, much like an HTML page, that can be viewed in a web browser. However, in addition to HTML tags, it can include a set of JSP tags and directives intermixed with Java code that extend the ability of the web page designer to incorporate dynamic content in a page. These additional features provide functionality such as displaying property values and using simple conditionals. Sun ONE Web Server 6.1 supports the JavaServer Pages (JSP) 1.2 API specification.

NOTE Ensure that the case of the URI your application requests for (for example, `/foo.JSP`) matches the canonical case of the file system path (for example, `C:\Program Files\WebServer\docs\foo.jsp`). This is necessary because the Sun ONE Web Server 6.1 Java web container currently performs case-sensitive pattern matches.

For information about creating JSPs, see Sun Microsystem's JavaServer Pages web site at:

`http://java.sun.com/products/jsp/index.html`

For information about developing servlets and JSPs for use with Sun ONE Web Server, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications.*

What the Server Needs to Run Servlets

Sun ONE Web Server includes the Java Development Kit (JDK) version 1.4.1_03. While in previous versions of the Web server, Java was configured server-wide, in the 6.1 release, you can configure Java per instance of the Web server.

You may use the JDK that is bundled with Sun ONE Web Server 6.1 or you may use a JDK of your choice, in which case you must specify a path to the JDK. For more information on how you can do this, see "Configuring JVM Settings" on page 281.

By default, Java is disabled when you install the Sun ONE Web Server. In order to enable servlets, you must first enable Java.

For information on how you can enable Java, see "Enabling and Disabling Java" on page 279.

Deploying Web Applications

The following sections describe how to deploy, edit, and delete web applications either manually by using the `wdeploy` command line utility, or through the user interface.

Using the server.xml File

Once deployed, your web applications are enabled by default. To disable a deployed web application manually you would need to modify the `server.xml` file as follows:

```
<VS>
<WEBAPP uri="/mywebapp" path="/webappdir" enabled = "false" >
</WEBAPP>

...

</VS>
```

If you inadvertently deploy or edit more than one web application with the same description, and one of them is disabled, the server will ignore `enabled = "false"` and continue with default setting of `enabled = "true"`.

For more information about the `server.xml` file, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications.*

You can deploy and edit web applications in two ways:

- Using the Administration Server Interface
- Using the Command Line Interface

Using the Administration Server Interface

Using Sun ONE Web Server 6.1, you can deploy, edit, delete, disable, and enable web applications for a specified virtual server.

Deploying Web Applications

You can access the Deploy Web Applications page by selecting Deploy Web Applications under the Web Applications tab of the Virtual Server Manager.

To deploy a web application, follow these steps:

1. Select Local Machine or Server Machine from the WAR File On drop-down list.

   Select Local Machine when uploading a WAR file to your server. Select Server Machine when the WAR file already resides there.

2. Enter the path on the local or server machine to the `WAR` file containing the web application in the field provided.

   On server machines enter the absolute path to the WAR file.

   On local machines you can browse the available paths. Clicking browse will bring up the File Upload window, allowing you to select the WAR file to upload to your server.

3. Enter the URI on the virtual server for the web application in the field provided.

4. Enter the absolute path to the directory on the server machine into which the contents of the WAR file will be extracted. If the directory does not exist, one will be created.

5. Click OK.

6. Click Apply.

7. Select Dynamic Reconfiguration for your web application to be deployed.

Editing Web Applications

You can edit, delete, disable, or enable an already deployed web application. Access the Edit Web Applications page by selecting Edit Web Applications under the Web Applications tab of the Virtual Server Manager.

To edit, delete, disable, or enable an already deployed web application, follow these steps:

1. Select the action you wish to perform from the drop-down list in the Action column next to the web application you are editing. Choose:
   - Edit to change the URI where the web application can be accessed.
   - Delete to delete the web application entry from the web applications file and delete the directory where the application is deployed.
   - Disable to make the web application inaccessible from the URI, but not delete it.
   - Enable to reactivate web applications that were previously disabled.

CAUTION Deleting a web application also deletes the directory the application is deployed in.

2. (Optional) Enter a new URI in the URI field if you are editing the web application.
3. Click OK.
4. Click Apply.
5. Select Dynamic Reconfiguration for your web application to be deployed.

Using the Command Line Interface

Before you can deploy a web application manually, you must make sure that the *server_root*`/bin/https/httpsadmin/bin` directory is in your path and that the `IWS_SERVER_HOME` environment variable is set to your *server_root* directory.

To deploy a virtual server web application:

You can use the `wdeploy` utility at the command line to deploy a WAR file into a virtual server web application environment:

```
wdeploy deploy -u <uri_path> -i <instance> -v <vs_id> [ [-V
<verboseLevel>]| [-q] ] [-n] [-d <directory>] <war_file>
```

To delete a virtual server web application:

```
wdeploy delete -u <uri_path> -i <instance> -v <vs_id> [ [-V
<verboseLevel>] | [-q] ] [-n] hard|soft
```

To list the web application URIs and directories for a virtual server:

```
wdeploy list -i <instance> -v <vs_id> [ [-V <verboseLevel>] | [ -q]
]
```

The command parameters have the following meanings:

uri_path	The URI prefix for the web application.
instance	The server instance name.
vs_id	The virtual server ID.
directory	(optional) The directory to which the application is deployed, or from which the application is deleted. If not specified for deployment, the application is deployed to the document root directory.
`hard | soft`	Specifies whether the directory and the `server.xml` entry are deleted (`hard`) or just the `server.xml` entry is deleted (`soft`).
war_file	The WAR file name
verboseLevel	The verbose level to display the log messages on console. The value can range from 0 to 4. The default value is 1. Note that in Sun ONE web Server 6.1, the `loglevel` attribute of the `LOG` element in `server.xml` is used in lieu of this element.
`-q`	(quiet) Sets the verbose level to zero. It is equivalent to the setting `-V 0`.
`-n`	prevents `wdeploy` from automatically sending the reconfigure command to the web server. For more information, see Using -n in the wdeploy Command.

CAUTION If you deploy a web application and do not specify a *directory*, the application is deployed to the document root directory. If you then delete the application using the `hard` parameter, the document root directory will be deleted.

When you execute the `wdeploy deploy` command, three things happen:

- A web application with the given *uri_path* and *directory* gets added to the `server.xml` file.

- The WAR file gets extracted at the target *directory*.
- The server is dynamically reconfigured to load the new web application.

For example:

```
wdeploy deploy -u /hello -i server.sun.com -v acme.com
-d /s1ws61/https-server.sun.com/acme.com/web-apps/hello
/s1ws61/plugins/servlets/examples/web-apps/HelloWorld/HelloWorld.wa
r
```

This utility results in the following `server.xml` entry:

```
<VS>
   <WEBAPP uri="/hello"
   dir="/s1ws61/https-server.sun.com/acme.com/webapps/hello"/>
</VS>
```

The `/s1ws61/https-server.sun.com/acme.com/web-apps/hello` directory has the following contents:

```
colors
index.jsp
META-INF
WEB-INF/
   web.xml
   /classes/
      HelloWorldServlet.class
      HelloWorldServlet.java
      SnoopServlet.class
      SnoopServlet.java
```

Using -n in the wdeploy Command

In Sun ONE Web Server 6.1, after deploying or deleting a web application, `wdeploy` dynamically reconfigures the server, causing the server to load or unload the web application that was deployed or deleted. Previously, you had to explicitly reconfigure the server in order for your changes to take effect by doing one of the following:

- Using the reconfig script
- Restarting the server
- Clicking the Apply link in the Administration User Interface.

Now a successful `wdeploy` command will automatically be enabled to service requests for a new web application, or to stop servicing requests for a deleted web application.

The `-n` option prevents `wdeploy` from automatically sending the reconfigure command to the web server. Use the `-n` option in your command when deploying or undeploying multiple web applications (in a script for example), and you want to reconfigure the server only once after the last web application is deployed.

Accessing Deployed Web Applications

After you have deployed an application, you can access it from a browser as follows:

`http://`*vs_urlhost*`[:`*vs_port*`]/`*uri_path*`/[`*index_page*`]`

The parts of the URL have the following meanings:

vs_urlhost	One of the `urlhosts` values for the virtual server.
vs_port	(optional) Only needed if the virtual server uses a non-default port.
uri_path	The same one you used to deploy the application. This is also the context path.
index_page	(optional) The page in the application that end users are meant to access first.

For example:

```
http://acme.com:80/hello/index.jsp
```

or:

```
http://acme.com/hello/
```

Return Values

The wdeploy option returns following exit values:

- `0`. Indicates that the `wdeploy` option was executed successfully.
- `1`. Indicates that an error occurred while executing the `wdeploy` option due to invalid command line arguments or invalid content of the configuration files.
- `2`. Indicates that the error is due to operating system settings. Either the specified directory doesn't exist or the file permission is not set.

Deploying Servlets and JSPs Not in Web Applications

You can deploy 4.*x* servlets and JSPs outside of web applications, but only in the default virtual server. For information, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*.

Configuring JVM Settings

You can configure attributes for the Java Virtual Machine (JVM) in the Java tab in the Server Manager.

For more information on these options, see the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*.

Deleting Version Files

The Delete Version Files page on the Java tab of the Server Manager allows you to delete the files that contain the version numbers for the JavaServer Pages class cache and the session data cache. This page has the following fields:

Clear Session Data

Deletes the SessionData directory, which stores persistent session information if the server uses the MMapSessionManager session manager.

Delete JSP ClassCache Files

Deletes the ClassCache directory, which caches information for JavaServer Pages (JSP). The default location of this directory is:

```
server_root/https-server_id/ClassCache/virtual_server_id/webapp_uri
/
```

When the server serves a JSP page, it creates a .java and a .class file associated with the JSP and stores them in the JSP class cache under the ClassCache directory.

The server uses two directories to cache information for JavaServer Pages (JSP) and servlets:

- `ClassCache`

  The server uses the following directory to cache information for JavaServer Pages (JSP):

server_root/https-*server_id*/ClassCache/*virtual_server_id*/*webapp_uri*/

When the server serves a JSP page, it creates a .java and a .class file associated with the JSP and stores them in the JSP class cache under the ClassCache directory.

- SessionData

  If the server uses the MMappedSessionManager session manager, it stores persistent session information in the SessionData directory.

Each cache has a version file containing a version number that the server uses to determine the structure of the directories and files in the caches. You can clean out the caches by simply deleting the version file.

When the server starts up, if it does not find the version files, it deletes the directory structure for the corresponding caches and re-creates the version files. Next time the server serves a JSP page, it recreates the JSP class cache. The next time the server serves a JSP page or servlet while using MMappedSessionManager session manager, it recreates the session data cache.

If a future upgrade of the server uses a different format for the caches, the server will check the number in the version file and clean up the caches if the version number is not correct.

Installing CGI Programs

This section discusses how to install CGI programs. The following topics are described:

- Overview of CGI
- Specifying a CGI Directory
- Specifying CGI as a File Type
- Downloading Executable Files

In addition, the following sections discuss how to install Windows-specific CGI programs:

- Installing Windows CGI Programs
- Installing Shell CGI Programs for Windows

Overview of CGI

Common Gateway Interface (CGI) programs can be defined with any number of programming languages. On a UNIX/Linux machine, you're likely to find CGI programs written as Bourne shell or Perl scripts.

NOTE Under UNIX/Linux, there are extra `CGIStub` processes running that the server uses to aid in CGI execution. These processes are created only during the first access to a CGI. Their number varies depending upon the CGI load on the server. Do not kill these CGIStub processes. They disappear when the server is stopped.

On a Windows computer, you might find CGI programs written in C++ or batch files. For Windows, CGI programs written in a Windows-based programming language such as Visual Basic use a different mechanism to operate with the server. They are called Windows CGI programs. See "Installing Windows CGI Programs" on page 359 for information about Windows CGI.

NOTE In order to run the command-line utilities, you need to manually set the `Path` variable to include *server_root*`/bin/https/bin`.

Regardless of the programming language, all CGI programs accept and return data in the same manner. For information about writing CGI programs, see the following sources of information:

- Sun ONE Web Server 6.1 *Programmer's Guide*
- *The Common Gateway Interface* at:

  `http://hoohoo.ncsa.uiuc.edu/cgi/overview.html`

- Articles about CGI available on the online documentation web site at:

  `http://docs.sun.com`

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

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

You can enable both options at the same time if desired.

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

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

NOTE By default, the file extensions for CGI programs are `.cgi`, `.exe` and `.bat`. However, you can change which extensions indicate CGI programs by modifying the MIME types file. You can do this by choosing the Server Preferences tab and clicking the MIME Types link.

Specifying a CGI Directory

To specify a CGI-only directory for a class of virtual servers, perform the following steps:

1. From the Class Manager, choose the Programs tab.

   The CGI Directory window appears.

2. In the URL Prefix field, type the URL prefix to use for this directory. That is, the text you type appears as the directory for the CGI programs in URLs.

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

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

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

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

4. Click OK.
5. Save and apply your changes.

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

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

Configuring Unique CGI Attributes for Each Software Virtual Server

To specify CGI attributes for a single virtual server, perform the following steps:

1. From the Class Manager, choose the Manager Virtual Servers button.
2. From the Virtual Server Manager, choose the Settings link.
3. In the CGI User text field, type the name of the user to execute CGI programs as.
4. In the CGI Grouptext field, type the name of the group to execute CGI programs as.
5. In the CGI Directory text field, type the directory to `chdir` to after `chroot` but before execution begins.
6. (UNIX only) In the CGI Nice text field, type an increment that determines the CGI program's priority relative to the server. Typically, the server is run with a nice value of 0 and the nice increment would be between 0 (the CGI program runs at same priority as server) and 19 (the CGI program runs at much lower priority than server). While it is possible to increase the priority of the CGI program above that of the server by specifying a nice increment of -1, this is not recommended.
7. In the Chroot Directory text field, type the directory to chroot to before execution begins.
8. Click OK.
9. Save and apply your changes.

Specifying CGI as a File Type

To specify CGI programs as a file type, perform the following steps:

1. From the Class Manager, choose the Programs tab.
2. Click the CGI File Type page.

   The CGI as a File Type window appears.
3. From the Editing picker, choose the resource you want this change to apply to.
4. Click the Yes radio button under Activate CGI as a File Type.
5. Click OK.
6. Save and apply your changes.

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

Downloading Executable Files

If you're using `.exe` as a CGI file type, you cannot download `.exe` files as executables.

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

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

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

`http://help.netscape.com/kb/server/960513-130.html`

Installing Windows CGI Programs

This section discusses how to install Windows CGI Programs. The following topics are included in this section:

Overview of Windows CGI Programs

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

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

Sun ONE Web Servers support the Windows CGI 1.3a informal specification, with the following differences:

- The following keywords have been added to the [CGI] section to support security methods:
  - HTTPS: its value is on or off, depending on whether the transaction is conducted through SSL.
  - HTTPS Keysize: when HTTPS is on, this value reports the number of bits in the session key used for encryption.
  - HTTPS Secret Keysize: when HTTPS is on, this value reports the number of bits used to generate the server's private key.
- The keyword Document Root in the [CGI] section might not refer to the expected document root because the server does not have a single document root. The directory returned in this variable is the root directory for the Windows CGI program.
- The keyword Server Admin in the [CGI] section is not supported.
- The keyword Authentication Realm in the [CGI] section is not supported.
- Forms sent with multi-part/form-data encoding are not supported.

Specifying a Windows CGI Directory

To specify a Windows CGI-only directory:

1. From the Class Manager, choose the Programs tab.
2. Click the WinCGI Directory link.

   The WinCGI Directory window appears.
3. In the URL Prefix text field, enter the URL prefix you want to use for this directory.

   That is, the text you type appears as the directory for the Windows CGI programs in URLs. For example, if you type `wcgi-programs` as the URL prefix, then all URLs to these Windows CGI programs have the following structure:

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

NOTE The URL prefix you specify can be different from the real Windows CGI directory you specify in Step 5.

4. Choose whether you want to enable script tracing.

   Click the Yes or No radio button under “Enable Script Tracing?”.

   CGI parameters are passed from the server to Windows CGI programs through files, which the server normally deletes after the Windows CGI program finishes execution. If you enable script tracing, these files are retained in a `/temp` directory or wherever the environment variables TMP and TEMP are pointing. Also, any window that the Windows CGI program brings up is shown when script tracing is enabled.
5. In the WinCGI Directory field, enter the location of the directory as an absolute path.

   Note that this directory doesn’t have to be under your document root. This is the reason that you need to specify a URL prefix in Step 3.
6. Click OK.
7. Save and apply your changes.

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

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

Specifying Windows CGI as a File Type

To specify a file extension for Windows CGI files, perform the following steps:

1. From the Server Manager, choose the Server Preferences tab.
2. Click the MIME Types link.

   The Global MIME Types window appears. For more information on the Global MIME Types, see "Choosing MIME Types" on page 175.
3. Add a new MIME type with the following settings:
   - Type: `type`
   - Content type: `magnus-internal/wincgi`.
   - File Suffix: Enter the file suffixes that you want the server to associate with Windows CGI. If you activated CGI, WinCGI, and shell CGI file types, you must specify a different suffix for each type of CGI. For example, you can't use the suffix `.exe` for both a CGI program and a shell CGI program. If you need to, you can edit the other MIME type fields on the page so that the suffixes are unique.
4. Click the New Type button.
5. Save and apply your changes.

Installing Shell CGI Programs for Windows

This section discusses how to install Shell CGI Programs for Windows. The following topics are included in this section:

Overview of Shell CGI Programs for Windows

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

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

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

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

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

Specifying a Shell CGI Directory (Windows)

To create a directory for your shell CGI files, perform the following steps:

1. Create the shell directory on your computer. This directory doesn't have to be a subdirectory of your document root directory.
2. From the Server Manager, choose the Class Manager link.
3. Next, choose Class Manager.

   The shell CGI Directory link is highlighted and the CGI window appears.
4. In the URL Prefix field, enter the URL prefix you want to associate with your shell CGI directory.

   For example, suppose you store all shell CGI files in a directory called `C:\docs\programs\cgi\shell-cgi`, but you want users to see the directory as `http://www.`*yourserver*`.com/shell/`. In this case, you would type `shell` as the URL prefix.
5. In the Shell CGI Directory field, enter the absolute path to the directory you created.

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

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

Specifying Shell CGI as a File Type (Windows)

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

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

To associate a file extension as a shell CGI file, perform the following steps:

1. Create the shell directory on your computer. This directory doesn't have to be a subdirectory of your document root directory.
2. From the Server Manager, choose Server Preferences.
3. Click the MIME Types link.

   The Global MIME Types window appears. For more information on the Global MIME Types, see "Choosing MIME Types" on page 175.
4. Add a new MIME type with these settings:
   - Type: `type`
   - Content type: `magnus-internal/shellcgi`.
   - File Suffix: Enter the file suffixes that you want the server to associate with shell CGI. If you activated CGI, WinCGI, and shell CGI file types, you must specify a different suffix for each type of CGI. For example, you can't use the suffix `.exe` for both a CGI program and a shell CGI program. If you need to, you can edit the other MIME type fields on the page so that the suffixes are unique.
5. Click the New Type button.

6. Save and apply your changes.

Using the Query Handler

NOTE The use of Query Handlers is outdated. Although Sun ONE Web Server and Netscape Navigator clients still support it, it is rarely used. It is much more common for people to use forms in their HTML pages to submit queries.

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

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

To set a query handler, perform the following steps:

1. From the Class Manager, choose the Programs tab.
2. Click the Query Handler link.

   The Query Handler window appears.
3. Use the Editing Picker to select the resource you want to set with a default query handler.

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

Chapter 16

Content Management

This chapter describes how you can configure and manage content for classes of virtual servers and virtual servers.

This chapter contains the following sections:

Setting the Primary Document Directory

The primary document directory (also called the document root) is the central directory where you store all the files you want to make available to remote clients.

When you add a class, you specify a document directory with an absolute path. If you do not use a variable as part of that path, the document root for every virtual server in the class will default to the same directory. You can then change them individually in the Class Manager.

Another approach is to use a variable when you set the path for the class. For example, you can use the `$id` variable to create a directory named with the virtual server id for every virtual server in the class. You can set the class' document root to be *class_doc_root* / $id. Using this path, if your class' document directory is `/sun/servers/docs/$id`, the default document directory for a virtual server `vs1` that belongs to the class is `/sun/servers/docs/vs1`.

For more information about the document directory and how it is used at the server instance, class, and virtual server level, see"Document Root" on page 313.

To change the primary document directory to use a different path or variable, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Primary Document Directory.
3. Enter an absolute directory path or a variable, or a path and variable combination next to the virtual server.

   If you include the variable $id at the end of your document root absolute path, every virtual server by default, will have a default document root of *class_doc_root* / *virtual_server_ID*. For example, if your class' document directory is `/sun/servers/docs/$id`, the default document directory for a virtual server `vs1` that belongs to the class is `/sun/servers/docs/vs1`.

   For more information about variables, see "Using Variables" on page 317.
4. Click OK.

For more information, see the online help for the Primary Document Directory page.

NOTE Typically, each virtual server has its own primary document directory.

Setting Additional Document Directories

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

If you set up an additional document directory without using variables, that directory will be set at the class level, and used by all virtual servers in the class.

If you want to set up additional document directories for individual virtual servers in the class, you must use variables so that the directory the URL prefix is mapped to is different for every virtual server.

To add an additional document directory, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Additional Document Directories.
3. Choose the URL prefix to map.

   Clients send this URL to the server when they want documents.
4. Specify the directory to map those URLs to.
5. If you want to, use an existing configuration style to specify how this directory should be configured.
6. Click OK.

For more information, see the online help for the Additional Document Directories page.

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

- `/manual`
- `/servlet`

You should restrict access to these directories so that users cannot write to them. A sample ACL would be:

```
deny (all) anyone;
allow (rxli) all;
allow (wd) privileged_user;
```

Customizing User Public Information Directories (UNIX/Linux)

Sometimes users want to maintain their own web pages. You can configure public information directories that let all the users on a server create home pages and other documents without your intervention.

You can only set these up for the entire class. There's no way to customize them on a per virtual server basis.

With this system, clients can access your server with a certain URL that the server recognizes as a public information directory. For example, suppose you choose the prefix ~ and the directory `public_html`. If a request comes in for `http://www.sun.com/~jdoe/aboutjane.html`, the server recognizes that `~jdoe` refers to a users' public information directory. It looks up `jdoe` in the system's user database and finds Jane's home directory. The server then looks at `~/jdoe/public_html/aboutjane.html`.

To configure your server to use public directories, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click User Document Directories.
3. Choose a user URL prefix.

   The usual prefix is ~ because the tilde character is the standard UNIX/Linux prefix for accessing a user's home directory.
4. Choose the subdirectory in the user's home directory where the server looks for HTML files.

   A typical directory is `public_html`.
5. Designate the password file.

   The server needs to know where to look for a file that lists users on your system. The server uses this file to determine valid user names and to find their home directories. If you use the system password file for this purpose, the server uses standard library calls to look up users. Alternatively, you can create another user file to look up users. You can specify that user file with an absolute path.

   Each line in the file should have this structure (the elements in the `/etc/passwd` file that aren't needed are indicated with `*`):

   ```
   username:*:*:groupid:*:homedir:*
   ```

6. Choose whether to load the password database at startup.

   For more information, see “Loading the Entire Password File on Startup” on page 371.

7. Choose whether to apply a configuration style.

8. Click OK.

For more information, see the online help for the User Document Directories page.

Another way to give users separate directories is to create a URL mapping to a central directory that all of your users can modify.

Restricting Content Publication

In some situations a system administrator may want to restrict what user accounts are able to publish content via user document directories. To restrict a user’s publishing, add a trailing slash to the user’s home directory path in the `/etc/passwd file`:

```
jdoe::1234:1234:John Doe:/home/jdoe:/bin/sh
```

becomes:

```
jdoe::1234:1234:John Doe:/home/jdoe/:/bin/sh
```

After you make this modification, Sun ONE Web Server will not serve pages from this user’s directory. The browser requesting the URI receives a “404 File Not Found” error and a 404 error will be logged to the web server access log. No error will be logged to the errors log.

If, at a later time, you decide to allow this user to publish content, remove the trailing slash from the `/etc/passwd` entry, then restart the web server.

Loading the Entire Password File on Startup

You also have the option of loading the entire password file on startup. If you choose this option, the server loads the password file into memory when it starts, making user lookups much faster. If you have a very large password file, however, this option can use too much memory.

Using Configuration Styles

You can apply a configuration style for the server to control access to directories from public information directories. This prevents users from creating symbolic links to information you do not want made public. For more information on configuration files, see Chapter 17, “Applying Configuration Styles”.

Enabling Remote File Manipulation

When you enable remote file manipulation, clients are able to upload files, delete files, create directories, remove directories, list the contents of a directory, and rename files on your server. The file `obj.conf` in the directory *`server_root`*`/https-`*`serve-id`*`/config` contains the commands that are activated when you enable remote file manipulation. By activating these commands, you allow remote browsers to change a server’s documents. You should use access control to restrict write access to these resources to prevent unauthorized tampering.

Note that enabling remote file manipulations should have no effect on using content management systems such as Microsoft Frontpage.

UNIX/Linux: You must have the correct permissions for your files or this function will not work; that is, the document root user must be the same as the server user.

To enable remote file manipulation, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Remote File Manipulation.
3. Choose to activate remote file manipulation.
4. Click OK.

For more information, see the online help for the Remote File Manipulation page.

Configuring Document Preferences

You use the Document Preferences page to set document preferences. This section discusses these topics:

- Setting the Document Preferences
- Entering an Index Filename

- Selecting Directory Indexing
- Specifying a Server Home Page
- Specifying a Default MIME Type

These settings are all configured for the class, not individual virtual servers.

Setting the Document Preferences

To set the document preferences, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Document Preferences.
3. Choose the appropriate field values, as discussed in the following sections.
4. Click OK.

The preferences you can set are discussed more fully in the sections that follow. For additional information, see the online help for the Document Preferences page.

Entering an Index Filename

If a document name is not specified in the URL the server automatically displays the index file. The default index files are `index.html` and `home.html`. If more than one index file is specified, the server looks in the order in which the names appear in this field until one is found. For example, if your index filenames are `index.html` and `home.html`, the server looks for `index.html` and if it doesn't find it looks for `home.html`.

Selecting Directory Indexing

A document directory will probably have several subdirectories. For example, there might be a directory called `products`, another called `people`, and so on. It's often helpful to let clients access an overview (or index) of these directories.

The server indexes directories by searching the directory for an index file called `index.html` or `home.html`, which is a file you create and maintain as an overview of the directory's contents. For more information, see the previous section, "Entering an Index Filename" on page 373. You can specify any file as an index file for a directory by naming it one of these default names, which means you can also use a CGI program as an index if CGI is activated.

If an index file isn't found, the server generates an index file that lists all the files in the document root.

CAUTION If your server is outside the firewall, turn off directory indexing to ensure that your directory structure and filenames are not accessible.

Specifying a Server Home Page

When end users first access the server, the first file they see is usually called a home page. Usually, this file has general information about your server and links to other documents.

By default, the server finds the index file specified in the Index Filename field in the Document Preferences page and uses that for the home page. However, you can also specify a file to use as the home page.

Specifying a Default MIME Type

When a document is sent to a client, the server includes a section that identifies the document's type, so the client can present the document in the right way. However, sometimes the server can't determine the proper type for the document because the document's extension is not defined for the server. In those cases, a default value is sent.

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

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

- `application/x-gzip`
- `audio/basic`

Configuring URL Forwarding

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

For example, if you forward `http://www.sun.com/info/movies` to a prefix `film.sun.com`, the URL `http://www.sun.com/info/movies` redirects to `http://film.sun.com/info/movies`.

You can use variables to map directories to new directories. For example, you can map `/new` to `/$docroot/new`. The mapping will go to the document root for the virtual server.

For more information about variables, see “Using Variables” on page 317.

Sometimes you may want to redirect requests for all the documents in one sub-directory to a specific URL. For example, if you had to remove a directory because it was causing too much traffic, or because the documents were no longer to be served for any reason, you could direct a request for any one the documents to a page explaining why the documents were no longer available. For example, a prefix on `/info/movies` could be redirected to `http://www.sun.com/explain.html`.

To configure URL forwarding, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click URL Forwarding.
3. Type the URL prefix you want to redirect, and whether you want to redirect it to another prefix or to a static URL.
4. Click OK.

For more information see the online help for the URL Forwarding page.

Customizing Error Responses

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

For example, you can change the way the server behaves when it gets an error for a specific directory. If a client tries to connect to a part of your server protected by access control, you might return an error file with information on how to get an account.

Before you can enable a custom error response, you must create the HTML file to send or the CGI program to run in response to an error. After you do this, enable the response in the Class Manager.

To enable a customized error response, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Error Responses.
3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.
4. For each error code you want to change, specify the absolute path to the file or CGI that contains the error response.
5. Click OK.

For more information see the online help for the Error Responses page.

Changing the Character Set

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

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

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

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

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

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

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

The following aliases are recognized for iso_8859-1:

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

To change the character set, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click International Characters.
3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.
4. Set the character set for all or part of the server.

   If you leave this field blank, the character set is set to NONE.

5. Click OK.

For more information, see the online help for the International Characters page.

Setting the Document Footer

You can specify a document footer, which can include the last-modified time, for all the documents in a certain section of the server. This footer works for all files except output of CGI scripts or parsed HTML (.shtml) files. If you need your document footer to appear on CGI-script output or parsed HTML files, enter your footer text into a separate file and add a line of code or another server-side include to append that file to the page's output.

To set the document footer, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Document Footer.
3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.

   If you choose a directory, the document footer applies only when the server receives a URL for that directory or any file in that directory.
4. Specify the type of files that you want to have include the footer.
5. Specify the date format.
6. Type any text you want to have appear in the footer.

   The maximum number of characters for a document footer is 765. If you want to include the date the document was last modified, type the string `:LASTMOD:`.
7. Click OK.

For more information see the online help for the Document Footer page.

Using htaccess

For information on using htaccess, see “Using .htaccess Files” on page 216.

Restricting Symbolic Links (UNIX/Linux)

You can limit the use of the file system links in your server. File system links are references to files stored in other directories or file systems. The reference makes the remote file as accessible as if it were in the current directory. There are two types of file system links:

- **Hard links**—A hard link is really two filenames that point to the same set of data blocks; the original file and the link are identical. For this reason, hard links cannot be on different file systems.
- **Symbolic (soft) links**—A symbolic link consists of two files, an original file that contains the data, and another that points to the original file. Symbolic links are more flexible than hard links. Symbolic links can be used across different file systems and can be linked to directories.

For more information about hard and symbolic links, see your UNIX/Linux system documentation.

File system links are an easy way to create pointers to documents outside of the primary document directory and anyone can create these links. For this reason you might be concerned that people might create pointers to sensitive files (for example, confidential documents or system password files).

To restrict symbolic links, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Symbolic Links.
3. Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.
4. Choose whether to enable soft and/or hard links and the directory to start from.
5. Click OK.

For more information, see the online help for the Symbolic Link page.

Setting up Server-Parsed HTML

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

To parse HTML, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Parse HTML.
3. Choose a resource for which the server will parse HTML.

   Choose Entire Server from the resource picker to apply your change to the whole class, or navigate to the document root for a specific virtual server, or to a specific directory or within a specific virtual server.

   If you choose a directory, the server will parse HTML only when the server receives a URL for that directory or any file in that directory.
4. Choose whether to activate server-parsed HTML.

   You can activate for HTML files but not the exec tag, or for HTML files and the exec tag, which allows HTML files to execute other programs on the server.
5. Choose which files to parse.

   You can choose whether to parse only files with the .shtml extension, or all HTML files, which slows performance. If you are using UNIX/Linux, you can also choose to parse UNIX/Linux files with the execute permission turned on, though that can be unreliable.
6. Click OK.

For more information on setting your server to accept parsed HTML, see the online help for the Parse HTML page.

For more information on using server-parsed HTML, see the Sun ONE Web Server 6.1 *Programmer's Guide.*

Setting Cache Control Directives

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

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

http://www.ietf.org/

To set cache control directives, follow these steps:

1. From the Class Manager, click the Content Management tab.
2. Click Cache Control Directives
3. Fill in the fields. Valid values for the response directives are as follows:
   - **Public.** The response is cachable by any cache. This is the default.
   - **Private.** The response is only cachable by a private (non-shared) cache.
   - **No Cache.** The response must not be cached anywhere.
   - **No Store.** The cache must not store the request or response anywhere in nonvolatile storage.
   - **Must Revalidate.** The cache entry must be revalidated from the originating server.
   - **Maximum Age (sec).** The client does not accept a response that has an age greater than this age.
4. Click OK.

For more information see the online help for the Cache Control Directives page.

Using Stronger Ciphers

For information on setting stronger ciphers, see “Setting Stronger Ciphers” on page 147.

Configuring the Server for Content Compression

Sun ONE Web Server 6.1 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 Web 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.

You can configure your server in two ways to handle compressed data:

- Configuring the Server to Serve Precompressed Content
- Configuring the Server to Compress Content on Demand

For information on enhancing the server's compression-handling capabilities, see Compression-related Changes in obj.conf.

Configuring the Server to Serve Precompressed Content

You can configure Sun ONE Web Server to generate and store pre-compressed versions of files in a specified directory. When configured, and only if an `Accept-encoding: gzip` header is received, all requests for files from a directory configured to serve precompressed content are redirected to requests for an equivalent compressed file from that directory if such a file exists. For example, if the Web server receives a request for `myfile.html`, and both `myfile.html` and `myfile.html.gz` exist, then those requests with an appropriate `Accept-encoding` header receive the compressed file.

To configure your server to serve precompressed content, perform the following steps:

1. From the Class Manager, click the Content Management tab.
2. Click Serve Precompressed Content.
3. Enter the following information:

- **Editing.** Select the resource from where precompressed content will be served from the drop-down list. If you choose a directory, the server will serve precompressed content only when the server receives a URL for that directory or any file in that directory.

  Click the Browse button to browse the primary document directory, or click the Wildcard button to specify a wildcard pattern. For information on using wildcard patterns, see Wildcards Used in the Resource Picker.

- **Activate Serving Precompressed Content?** Allows you to instruct the server to serve precompressed content for the selected resource.

- **Check Age.** Specify whether to check if the compressed version is older than the non-compressed version. Possible values are `yes` and `no`.

  If set to `yes`, then the compressed version, if it is older than the non-compressed version, will not be selected.

  If set to `no`, then the compressed version, even if it is older than the non-compressed version, will always be selected.

  By default, the value is set to `yes`.

- **Vary Header.** Specifies whether to use 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`.

4. Click OK.

Configuring the Server to Compress Content on Demand

You can also configure the Sun ONE Web Server 6.1 to compresses transmission data on the fly. A dynamically generated HTML page doesn't exist until a user asks for it. This is particularly useful for e-commerce-based Web applications and database-driven sites.

To configure your server to compress content on demand, perform the following steps:

1. From the Class Manager, click the Content Management tab.

2. Click Compress Content on Demand.
3. Enter the following information:
   - **Editing.** Select the resource from where compressed content will be served dynamically on demand from the drop-down list. If you choose a directory, the server will serve compressed content only when the server receives a URL for that directory or any file in that directory.

     Click the Browse button to browse the primary document directory, or click the Wildcard button to specify a wildcard pattern. For information on using wildcard patterns, see Wildcards Used in the Resource Picker.
   - **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.
4. Click OK.

Compression-related Changes in obj.conf

When compression is enabled in the server, an entry gets added to the `obj.conf` file. A sample entry is shown below:

```
Output fn="insert-filter" filter="http-compression" type="text/*"
```

To restrict compression to documents of a particular type only, or to exclude browsers that don't work well with compressed content, you would need to edit the `obj.conf` file. For more information on what you need to do to accomplish this, see the Sun ONE Web Server 6.1 *NSAPI Programmer's Guide*.

Applying Configuration Styles

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

This chapter includes the following sections:

Creating a Configuration Style

To create a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click the New Style link.
4. Type the name you want to give the configuration style. Click OK.

   Sun ONE Web Server displays the Edit a Style page.

5. From the drop-down list, choose a configuration style to edit and click Edit this Style.
6. From the list of links available, click the category you want to configure for your style.

   You can configure the information listed in Table 17-1.
7. Fill out the form that appears, and click OK.
8. Repeat step 4 and step 5 to make any other configuration changes to the configuration style. Click OK.

When you choose a style to edit, your Resource Picker lists configuration styles instead of other resources. After you have finished editing a style, click OK and Save and Apply. The Resource Picker exits the styles mode. You can also choose to exit the styles mode by choosing Exit styles mode from the Resource Picker. For more information about the Resource Picker, see "Using the Resource Picker" on page 41 of Chapter 1, "Introduction to Sun ONE Web Server".

Table 17-1 Configuration Style Categories

Category	Description
CGI file type	Allows you to activate CGI as a file type. For more information about CGIs, see "Installing CGI Programs," on page 355 in Chapter 15, "Extending Your Server With Programs".
Character Set	Allows you to change the character set for a resource. For more information about character sets, see "Changing the Character Set," on page 376 in Chapter 16, "Content Management".
Default Query Handler	Allows you to set a default query handler for a server resource. For more information about query handling, see "Using the Query Handler," on page 365 in Chapter 15, "Extending Your Server With Programs".
Document Footer	Allows you to add a document footer to a server resource. For more information, see "Setting the Document Footer" on page 378 in Chapter 16, "Content Management".
`.htaccess` Configuration	Allows you to give people a subset of configuration options without giving them access to the Server Manager. For more information about access control, see Chapter 9, "Controlling Access to Your Server".
Require Stronger Security	Allows you to specify key size restrictions, or to reject access with a specific file.
Error Responses	Allows you to customize the error responses that clients see when they encounter an error from your server.

Table 17-1 Configuration Style Categories *(Continued)*

Category	Description
Log preferences	Allows you to set preferences for access logs. For more information about log preferences, see "Setting Access Log Preferences," on page 242 in Chapter 10, "Using Log Files".
Remote File Manipulation	Allows you to activate the file manipulation commands which allow remote browsers to change your server's documents. For more information, see "Enabling Remote File Manipulation" on page 372 in Chapter 16, "Content Management".
Server Parsed HTML	Allows you to specify whether the server parses files before they are sent to the client. For more information, see the Sun ONE Web Server 6.1 *Programmer's Guide*.
Serve Precompressed Content	Allows you to specify whether the server sends a precompressed version of the file. For more information, see "Configuring the Server to Serve Precompressed Content" on page 382 in Chapter 16, "Content Management".
Compress Content on Demand	Allows you to specify whether the server dynamically compresses content before it is sent to the client. For more information, see "Configuring the Server to Compress Content on Demand" on page 383 in Chapter 16, "Content Management".
Symbolic links (UNIX/Linux)	Allows you to limit the use of filesystem links in your server. For more information, see "Restricting Symbolic Links (UNIX/Linux)" on page 379 in Chapter 16, "Content Management".

For more information, see the New Style page in the online help.

Assigning a Configuration Style

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

To assign a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.

3. Click the Assign Style link.
4. Enter the prefix of the URL to which you are applying this configuration style.

   If you choose a directory inside the document root, only enter the path after the document root. If you enter `/*` after the directory, you apply the configuration style to all of the directory's contents.
5. Select the configuration style you want to apply.

   To remove any configuration style previously applied to the resource, apply the None configuration style. Click OK.

For more information, see the Assign a Style page in the online help.

Listing Configuration Style Assignments

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

To list the configuration style assignments, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click the List Assignments link.

   Sun ONE Web Server displays the List Assignments page, showing the configuration styles you applied to server resources.
4. To edit a configuration style assignment, click the Edit link next to the configuration style name.

For more information, see the List Assignments page in the online help.

Editing a Configuration Style

To edit a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click the Edit Style link.

4. Select the configuration style you want to edit and click the “Edit this style” button.
5. From the list of links available, click the category you want to configure for your style.

   For more information on these categories, see the section “Creating a Configuration Style” on page 385.
6. Fill out the form that appears, and then click OK.
7. Repeat Step 4 and Step 5 to make any other changes to the configuration style. Click OK.

When you choose a style to edit, your Resource Picker lists configuration styles instead of other resources. After you have finished editing a style, click OK and Save and Apply. The Resource Picker exits the styles mode. You can also choose to exit the styles mode by choosing Exit styles mode from the Resource Picker. For more information about the Resource Picker, see “Using the Resource Picker” on page 41 of Chapter 1, “Introduction to Sun ONE Web Server”.

For more information, see the Edit Style page in the online help.

Removing a Configuration Style

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

To remove a configuration style, perform the following steps:

1. Access the Class Manager.
2. Choose the Styles tab.
3. Click List Assignments link.
4. Select Edit Style Assignment you want to remove.
5. Click Remove this Assignment.

For more information, see the Remove Style page in the online help.

Chapter 18

Using Search

Sun ONE Web Server 6.1 includes a search feature that allows users to search documents on the server and display results on a web page. Server administrators create the indexes of documents against which users will search (called collections), and can customize the search interface to meet the needs of their users.

This chapter includes the following sections:

About Search

The search feature is installed with other web components during the installation of Sun ONE Web Server. Search is configured and managed at the virtual server level instead of the server instance level, as it was with Sun ONE Web Server 6.0.

The Search tab in the Virtual Server Manager is used to configure search for each virtual server. From this tab you can:

- Enable and disable the search feature
- Create, modify, delete, and reindex search collections
- Create, modify, and remove scheduled maintenance tasks for search collections

Information obtained from the administrative interface is stored in the `<server-root>/config/server.xml` file, where it is mapped within the VS element.

Server administrators can customize the search query and search results pages. This might include rebranding the pages with a corporate logo, or changing the way search results appear. In previous releases this was accomplished through the use of pattern files. Pattern files are not supported in Sun ONE Web Server 6.1. Instead, customizing is now performed using a set of JSP tag libraries included with the product. These libraries provide functionality similar to that provided by the pattern files. For more information about customizing the search interface, see Customizing Search Pages.

There is no global “on” or “off” functionality for search as there was in previous releases. Instead, a default search web application is provided and then enabled or disabled on a specific virtual server. This search application provides the basic web pages used to query collections and view results. The search application includes sample JSPs that demonstrate how to use the search tag libraries to build customized search interfaces.

CAUTION Unlike Sun ONE Web Server 6.0, version 6.1 does not provide access checking on search results. Due to the number of potential security models and realms, it is impossible to perform security checks and filter results from within the search application. It is the responsibility of the server administrator to ensure that appropriate security mechanisms are in place to protect content.

Sun ONE Web Server 6.1 provides support for multiple document search. Documents with different formats (such as HTML, ASCII, and PDF) can be indexed and searched against.

NOTE Sun ONE Web Server 6.1 does not support search on multiple document formats on the Linux platform.

The search engine used in previous releases has been replaced in Sun ONE Web Server 6.1 with a new search engine. Therefore, when you migrate from a previous release of the Web server to Sun ONE Web Server 6.1, your existing search collections and indexes are not migrated.

Enabling the Search Application for a Virtual Server

Search is enabled for a virtual server by enabling the search application included with Sun ONE Web Server. The administrative interface is used to enable search.

NOTE The Java web container must be enabled for search to be enabled.

After ensuring that Java is enabled for the virtual server class that contains the virtual server you want to configure, enable search by performing the following steps:

1. Select the virtual server for which you want to enable search, and click the Manage button.
2. Select the Search tab and then click the Search Configuration link.
3. Enter the following information:
   - **Max Hits.** Specify the maximum results retrieved in a search query.
   - **URI.** If you plan to use a custom search application, enter the URI; if you are using the default search application, you don't need to specify a value here.
   - **Path.** If you plan to use a custom search application, enter the path; if you are using the default search application, you don't need to specify a value here.

- **Enabled.** Check this to enable the default search application.

4. Click OK.

Disabling the Search Application for a Virtual Server

Search is disabled for a virtual server by disabling the search application included with Sun ONE Web Server. The administrative interface is used to disable search.

To disable search for a virtual server, perform the following steps:

1. Select the virtual server for which you want to disable search, and click the Manage button.
2. Select the Search tab and then click the Search Configuration link.
3. Uncheck the Enabled checkbox.
4. Click OK.

About Search Collections

Searches require a database of searchable data against which users will search. Server administrators create this database, called a collection, which indexes and stores information about documents on the server. Once the server administrator indexes all or some of a server's documents, information such as title, creation date, and author is available for searching.

Please note the following about collections:

- Collections are specific to the virtual server being administered
- Only documents visible from the virtual server are presented in the administrative interface and available to be indexed
- There is no limit to the number of collections that can exist on your server
- A single search collection can only contain files that are located under one parent directory on the file system
- Documents with different formats (such as HTML, ASCII, and PDF) can be indexed and searched against

- Documents in a search collection are not specific to any one character encoding, which means that a search collection can be associated with multiple encodings
- Information about collections is stored in the VS element in `server.xml`

This section includes the following topics:

Creating a Collection

Collections are created and managed from the administrative interface. You create a new collection by specifying the documents to be indexed.

To create a new collection, perform the following steps:

1. Select the virtual server in which you want to create a collection, and click the Manage button.
2. Select the Search tab and then click the Create Collection link.
3. Enter the following information:

- **Directory to Index.** From the drop-down list, select the directory from which documents will be indexed into the collection. Only the directories visible from this virtual server will be listed.

  To view the contents of the directory, click View. If the selected directory has subdirectories, these are listed out in the "View *directory_name*" page. To select a directory to index, click index. To view a directory, click on the folder.

  In order to add a directory to the list of indexable directories, you must first create an additional document directory. For more information, see Setting Additional Document Directories.

- **Collection Name.** Enter a name for the collection.
- **Display Name.** (Optional) This will appear as the collection name in the search query page. If you don't specify a display name, the collection name serves as the display name.
- **Description.** (Optional) Enter text that describes the new collection.
- **Include Subdirectories?** If you select No, documents within the subdirectories of the selected directory will not be indexed. The default is Yes.
- **Pattern.** Specify a wildcard to select the files to be indexed. For more information on wildcards, see Wildcards Used in the Resource Picker.

CAUTION Use the wildcard pattern judiciously to ensure that only specific files are indexed. For example, specifying *.* might cause even executables and perl scripts to be indexed.

- **Default Encoding.** Specify the character encoding for the documents to be indexed. The default is "ISO-8859-1." The indexing engine tries to determine the encoding of HTML documents from the embedded meta tag. If this is not specified, the default encoding is used.

  Documents in a collections are not restricted to a single language/encoding. Every time documents are added, only a single encoding can be specified; however, the next time you add documents to the collection, you can select a different default encoding.

4. Click OK.

This creates a new collection by the specified name in the following location:

<instance-root>/collections/*<vs-id>*/*<collection-name>*

It also creates an appropriate SEARCHCOLLECTION entry in the server.xml file.

Configuring a Collection

After a collection has been created, you can modify some of its settings. These settings are stored in the server.xml file. When you reconfigure a collection, the server.xml file is updated to reflect your changes.

You should avoid making unnecessary changes to collection settings.

To reconfigure an existing collection, perform the following steps:

1. Select the virtual server that contains the collection you want to configure, and click the Manage button.
2. Select the Search tab and then click the Configure Collection link.
3. From the Collection drop-down list, select the collection you want to configure and click Go.
4. You can edit the following information for the collection you selected:
   - **Display name.** (Optional) This will appear as the new collection name in the search query page.
   - **Description.** (Optional) Edit the text description of the collection.
   - **Document URI.** Edit the URI for the document root for the search collection.

NOTE Do not change the Document URI unless you have changed the URI mapping for the document root from the Additional Document Directories page. For more information, see Setting Additional Document Directories.

   - **Enabled.** Select Yes to enable. If you select No, the collection will not appear on the search query page.
5. Click OK

This reconfigures the collection and modifies the appropriate SEARCHCOLLECTION entry in the server.xml file.

Updating a Collection

You can add or remove files after a collection has been created. Documents can be added only from under the directory that was specified during collection creation. If you are removing documents, only the entries for the files and their metadata are removed from the collection. The actual files themselves are not removed from the file system.

To update a collection, perform the following steps:

1. Select the virtual server that contains the collection you want to update, and click the Manage button.
2. Select the Search tab and then click the Update Collection link.
3. From the Collection drop-down list, select the collection you want to update.
4. Docs
5. You can update the following information for the collection you selected:

    - **Include subdirectories?** If you select No, documents within the subdirectories of the selected directory will not be indexed. The default is Yes.

NOTE **Include Subdirectories?** has a bearing on only adding documents.

    - **Pattern.** Specify a wildcard to select the files to be indexed or removed from the collection. For more information on wildcards, see Wildcards Used in the Resource Picker.

CAUTION While adding documents, use the wildcard pattern judiciously to ensure that only specific files are indexed. For example, specifying *.* might cause even executables and perl scripts to be indexed.

    - **Default Encoding.** Specify the character encoding for the documents to be indexed. The default is “ISO-8859-1.” The indexing engine tries to determine the encoding of HTML documents from the embedded meta tag. If this is not specified, the default encoding is used.

Documents in a collections are not restricted to a single language/encoding. Every time documents are added, only a single encoding can be specified; however, the next time you add documents to the collection, you can select a different default encoding.

6. Click Add Documents to add documents to the index, or Remove Documents to remove the appropriate index entries.

NOTE You can add documents only if they are located in the directory you specified when you created the collection.

Removing a Collection

You can remove a collection after it has been created. When a collection is deleted, it is no longer visible to users on the search query page, and all configuration and index files associated with the collection are deleted. The actual documents that formed the collection are not deleted from the file system, just their index entries in the collection are deleted.

To remove a collection, perform the following steps:

1. Select the virtual server that contains the collection you want to remove, and click the Manage button.
2. Select the Search tab and then click the Maintain Collection link.
3. From the Collection drop-down list, select the collection you want to remove.
4. Click the Remove Collection button.

NOTE When a collection is removed, the maintenance scheduled for the collection is also removed. For information about scheduled maintenace, see Adding Scheduled Collection Maintenance.

NOTE Do not use your local file manager to remove collections because doing so will not update the corresponding configuration files.

Maintaining a Collection

Periodically, you may want to maintain your collections. These tasks may not be necessary unless you index and update collections frequently. You can:

- Reindex a collection
- Update a collection

Reindexing a Collection

You can reindex a collection after it has been created. If any documents are modified after the collection was created, the collection is reindexed. Reindexing a collection does not index any new content into the collection, but rather updates the existing contents of the collection. If index entries exist for documents that are no longer present in the server file system, those entries will be removed.

To reindex a collection, perform the following steps:

1. Select the virtual server that contains the collection you want to reindex, and click the Manage button.
2. Select the Search tab and then click the Maintain Collection link.
3. From the Collection drop-down list, select the collection you want to reindex.
4. Click the Reindex button.

Adding Scheduled Collection Maintenance

You can schedule maintenance tasks to be performed on collections at regular intervals. The tasks that can be scheduled are reindexing and updating. The administrative interface is used to schedule the tasks for a specific collection. You can specify the:

- Task to perform (reindexing or updating)
- Time of day to perform the task
- Day(s) of the week to perform the task

To add regular maintenance of a collection, perform the following steps:

1. Select the collection you want to schedule maintenance for and click the Add Scheduled Maintenance link.

2. Enter the following information:

   - **Task.** Select the task you want to automate. The choices are reindex and update.

     If you select Update, you must enter the following information:

     - **Recurse Subdirectories?** If you select No, documents within the subdirectoriesof the selected directory will not be indexed. The default is Yes.

   - **Pattern.** Specify a wildcard to select the files to be indexed. For more information on wildcards, see Wildcards Used in the Resource Picker.

CAUTION Use the wildcard pattern judiciously to ensure that only specific files are indexed. For example, specifying *.* might cause even executables and perl scripts to be indexed.

   - **Default Encoding.** Specify the character encoding for the documents to be indexed. The default is "ISO-8859-1." The indexing engine tries to determine the encoding of HTML documents from the embedded meta tag. If this is not specified, the default encoding is used.

     Documents in a collections are not restricted to a single language/encoding. Every time documents are added, only a single encoding can be specified; however, the next time you add documents to the collection, you can select a different default encoding.

   - **Scheduled Time.** (Required) Specify the time of day, in the HH:MM format, when you want the scheduled maintenance to run. For example, you might want to scheduled maintenance to run at the end of the day when it is likely that the documents in the collection have been modified.

   - **Schedule day(s) of week.** (Required) Check one or more of the checkboxes to specify the day or days of the week when the scheduled maintenance will run.

3. Click OK.

NOTE UNIX/Linux users must restart the cron control process after adding scheduled maintenance, in order for their changes to take effect.

Editing Scheduled Collection Maintenance

If your requirements change, you can change the properties of the scheduled maintenance for a collection. You might for example, decide to reschedule maintenance keeping in mind the time when your site is most likely to be updated.

To change the scheduled maintenance for a collection, perform the following steps:

1. From the Collection drop-down list, select the collection for which you want to reschedule maintenance.
2. Select the task you want to reconfigure, and enter the necessary information. For more details, see the Edit Scheduled Collection page in the online help.
3. Click OK.

NOTE When a collection is removed, the maintenance scheduled for the collection is also removed.

NOTE UNIX/Linux users must restart the cron control process after reconfiguring scheduled maintenance, in order for their changes to take effect.

Removing Scheduled Collection Maintenance

You can cancel scheduled maintenance of a collection if it is no longer needed.

To cancel scheduled maintenance, perform the following steps:

1. From the Collection drop-down list, select the collection for which you want to remove maintenance.
2. Select the task you want to for which you want to remove scheduled maintenance: Reindex or Update. If a task is scheduled the details are now displayed.
3. For an Update task, check the Delete checkbox next to the task you want to remove.
4. Click OK.

NOTE UNIX/Linux users must restart the cron control process after removing scheduled maintenance, in order for their changes to take effect.

Performing a Search

Users are primarily concerned with asking questions of the data in the search collections, and getting a list of documents in return. The search web application installed with Sun ONE Web Server provides default search query and search results pages. These pages can be used as they are, or customized using a set of JSP tags as described in Customizing Search Pages.

Users search against collections that have been created by the server administrator. They can:

- Input a set of keywords and optional query operators on which to search
- Search only collections that are visible to the virtual server
- Search against a single collection, or across a set of collections visible to the virtual server

Server administrators must provide users with the URL needed to access the search query page for a virtual server.

CAUTION Unlike Sun ONE Web Server 6.0, version 6.1 does not provide access checking on search results. Due to the number of potential security models and realms, it is impossible to perform security checks and filter results from within the search application. It is the responsibility of the server administrator to ensure that appropriate security mechanisms are in place to protect content.

The Search Page

The default URL end-users can use to access search functionality is:

`http://<server-instance>:port number/search`

Example:

`http://plaza:8080/search`

When the end-user invokes this URL, the Search page, which is a Java web application, is launched.

The following figure shows the default Search interface:

The Default Sun ONE Web Server Search Page

You can customize this page using a set of JSP tags as described in “Customizing Search Pages.”

Making a Query

A search query page is used to search against a collection. Users input a set of keywords and optional query operators, and then receive results on a web page displayed in their browser. The results page contains links to documents on the server that match the search criteria.

NOTE Server administrators can customize this search query page, as described in “Customizing Search Pages.”

To make a query, perform the following steps:

1. Access the Search web application by entering its URL in the Location bar of your browser, in the following format:

   `http://<server-instance>:port number/search`

2. In the search query page that appears, check the checkbox representing the collection you want to search in the "Search in" field.

3. Type in a few words that describe your query and hit the 'enter' key (or click on the Search button) for a list of relevant web pages.

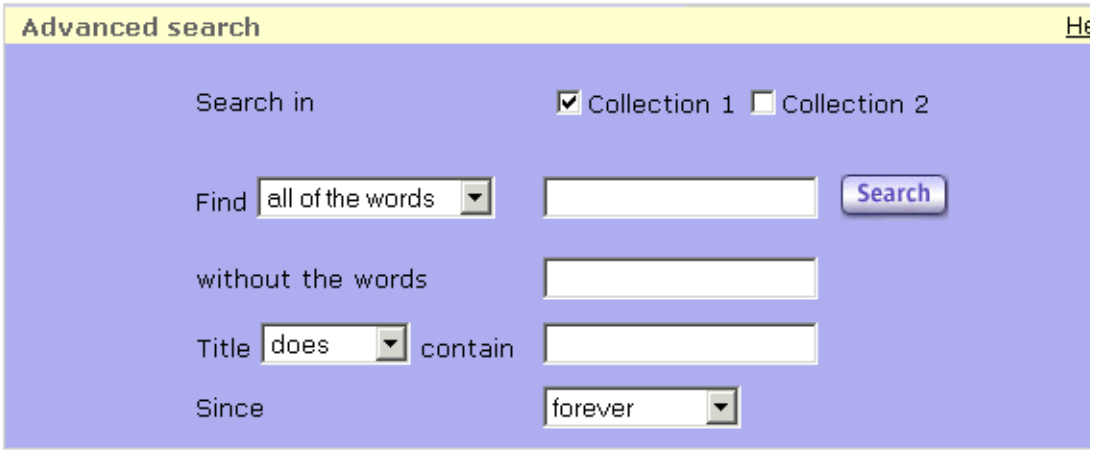
For a more fine-tuned search, you can use the search parameters provided in the Advanced Search page, described in the following section.

Advanced Search

Users can increase the accuracy of their searches by adding operators that fine-tune their keywords. These options can be selected from the Advanced Search page.

The following figure shows the advanced search page:

The Advanced Search Page



To make an advanced search query, perform the following steps:

1. Access the Search web application by entering its URL in the Location bar of your browser, in the following format:

   `http://<server-instance>:port number/search`

2. Click the Advanced link.
3. Enter any or all of the following information:
   - **Search in.** Select the collection you want to search.
   - **Find.** Four options are supported:
     - **All of the words.** Finds pages that include all the key words specified in Find.
     - **Any of the words.** Finds pages that include any of the key words specified in Find.
     - **The exact phrase.** Finds pages that match the exact phrase used in Find.
     - **Passage search.** Highlights the passage containing the keyword or words in the retrieved pages.

- **Without the words.** The search will exclude Web pages that contain the specified words.
- **Title “does/does not“ contain.** Restrict the search to pages with titles that include the specified key words.
- **Since.** Restrict the search operation to Web pages indexed in the selected time period.

Viewing Search Results

Search results are displayed in the user’s browser on a web page that contains HTML hyperlinks to documents on the server that match the search criteria. Each page displays 10 records (hits) by default, which are sorted in descending order based on relevance. Each record lists information such as file name, size, date of creation, and so on. The matched words are also highlighted.

NOTE Server administrators can customize this search results page, as described in Customizing Search Pages.

Customizing Search Pages

Sun ONE Web Server includes a default search application that provides basic search query and search results pages. These web pages can be used as is, or customized to meet your specific needs. Such customizing might be as simple as rebranding the web pages with a different logo, or as complex as changing the order in which search results are displayed.

Pattern files are no longer used to customize the search interface, as they were in Sun ONE Web Server 6.0. Instead, customizing is now done using a set of JSP tag libraries included with Sun ONE Web Server 6.1. The default search application provides sample JSPs that demonstrate how to use the search tag libraries to build customized search interfaces. You can take a look at the default search application located at `/bin/https/webapps/search` as a sample application that illustrates the use of customizable search tags.

The default search interface consists of four main components: header, footer, query form, and results.

These basic elements can be easily customized simply by changing the values of the attributes of the tags. More detailed customizing can be accomplished using the tag libraries.

This section includes the following topics:

Search Interface Components

The Search interface consists of the following components:

Header

The header includes a logo, title, and a short description.

Footer

The footer contains copyright information.

Form

The query form contains a set of check boxes representing search collections, a query input box, and submit and Help buttons.

Results

The results are listed by default in 10 records per page. For each record, information such as the title, a passage, size, date of creation, and URL are displayed. A passage is a short fragment of the page with matched words highlighted.

Customizing the Search Query Page

The query form contains a list of checkboxes for search collections, a query input box, and submit button. The form is created using the `<s1ws:form>` tag along with `<collElem>`, `<queryBox>`, and `<submitButton>` tags with default values:

```
<s1ws:form>
   <s1ws:collElem>
   <s1ws:queryBox> <s1ws:submitButton>
</s1ws:form>
```

The query form can be placed anywhere in a page, in the middle, on a side bar, and so on. It can also be displayed in different formats such as with a cross bar where the collection select box, the query string input box, and the Submit button are lined up horizontally, or in a block where the collections appear as checkboxes, and the query input box and Submit button are placed underneath.

The following examples show how the `<searchForm>` set of tags may be used to create query forms in different formats.

In a horizontal bar

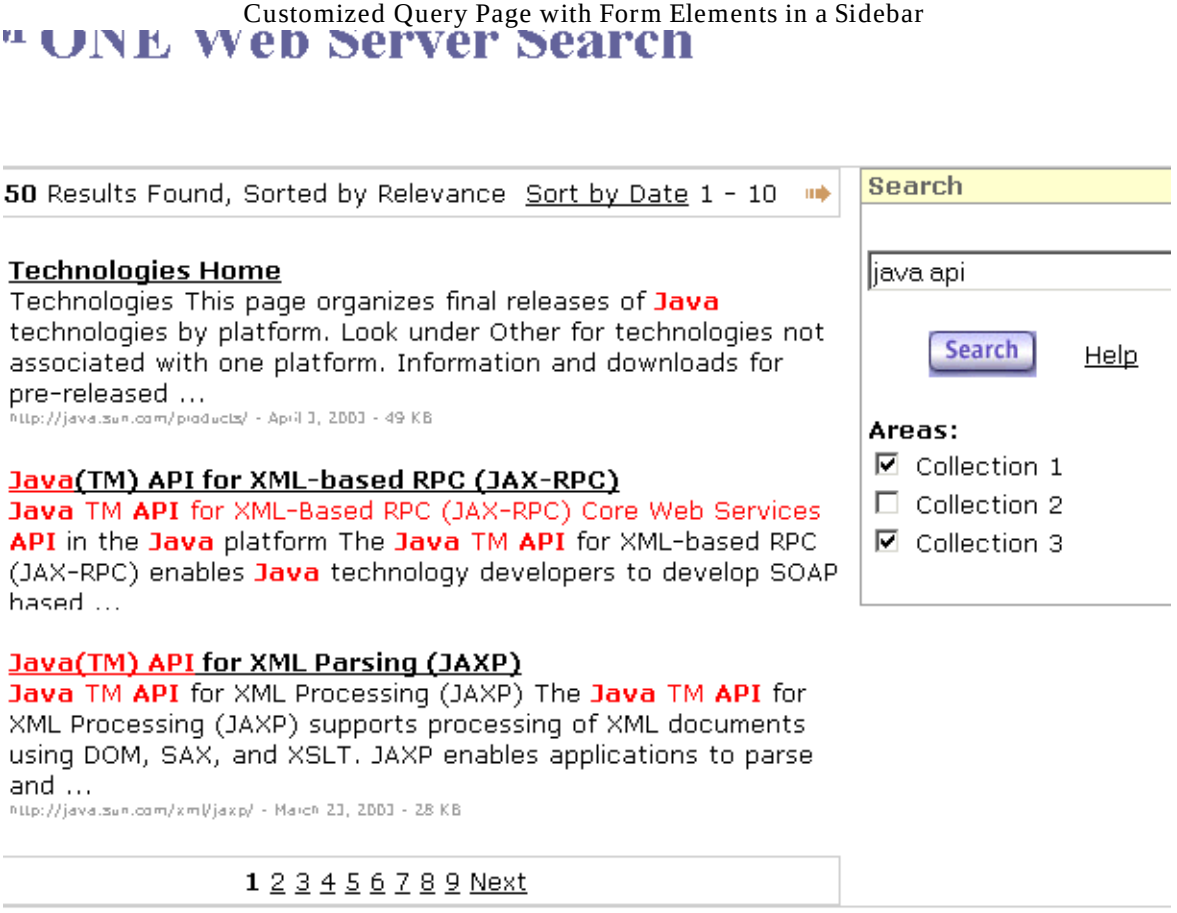
The sample code below would create a form with a select box of all collections, a query input box and a submission button all in one row.

```
<s1ws:form>
   <table cellspacing="0" cellpadding="3" border="0">
   <tr class="navBar">
      <td class="navBar"><s1ws:collElem type=”select”></td>
      <td class="navBar">
         <s1ws:querybox size="30">
         <s1ws:submitButton class="navBar" style="padding: 0px;
margin: 0px; width: 50px">
      </td>
   </tr>
   </table>
</s1ws:form>
```

In a Sidebar Block

You can create a form block in which form elements are arranged in a sidebar, and has the title "Search", which uses the same format as other items on the sidebar. The effect of such an arrangement is as shown in the following figure:

Customized Query Page with Form Elements in a Sidebar



In the sample code given below, the form body contains three checkboxes arranged in one column listing the available search collections. The query input box and the Submit button are placed underneath:

```
<s1ws:searchForm>
```

```
   <table>
<!--... other sidebar items ... -->
   <tr class="Title"><td>Search</td></tr>
   <tr class="Body">
      <td>
      <table cellspacing="0" cellpadding="3" border="0">
      <tr class="formBlock">
         <td class="formBlock"> <s1ws:collElem type="checkbox"
cols="1" values="1,0,1,0" /> </td>
      </tr>
      <tr class="formBlock">
         <td class="formBlock"> <s1ws:querybox size="15"
maxlength="50"> </td>
      </tr>
      <tr class="formBlock">
         <td class="formBlock"> <s1ws:submitButton class="navBar"
style="padding: 0px; margin: 0px; width: 50px"> </td>
      </tr>
      </table>
      </td>
   </tr>
   </table>
</s1ws:searchForm>
```

Customizing the Search Results Page

Search results are generated as follows:

- The `<formAction>` tag retrieves values from all of the form elements and conducts basic validations.
- The `<search>` tag, the `<resultIteration>` tag and other tags occur inside the `<formAction>` tag and have access to the values of all of the form elements.
- The `<search>` tag executes the search with the query string and collections from the `<formAction>` and saves the search results in `pageContext`.

- The <resultIteration> tag then retrieves and iterates through the result set.

You can customize the search results page simply by changing the attribute values of the tags.

The following sample code starts with a title bar, and then displays a number of records as specified, and finally, a navigation bar. The title bar contains the query string used in the search along with the range of total records returned, for example, 1 – 10. For each record, the records section shows the title with a link to the file, up to three passages with keywords highlighted, the URL, the date of creation, and the size of the document.

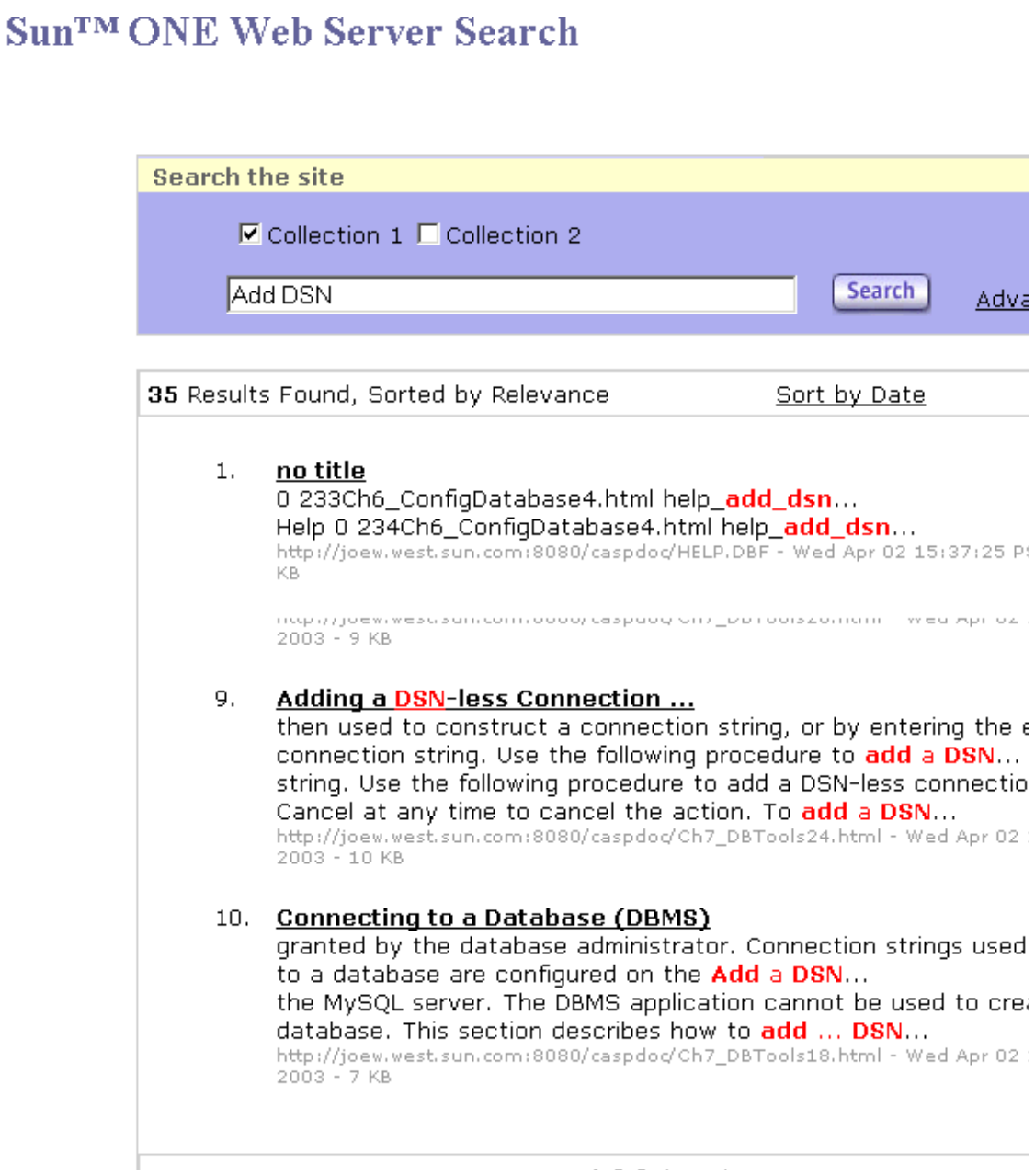
At the end of the section, the navigation bar provides links to the previous and next pages, as well as direct links to eight additional pages before and after the current page.

```
<s1ws:formAction />

<s1ws:formSubmission success="true" >

   <s1ws:search scope="page" />

   <!--search results-->

   (...html omitted...)

      <s1ws:resultStat formId="test" type="total" /></b> Results
Found, Sorted by Relevance</span></td><td>

      <span class="body"><a href="/search/search.jsp?">Sort by
Date</a></span></td>

      <td align="right"><span class="body">

      <s1ws:resultNav formId="test" type="previous" caption="<img
border=0 src=\"images/arrow-left.gif\" alt=\"Previous\">" />

       <s1ws:resultStat formId="test" type="range" />

       <s1ws:resultNav formId="test" type="next" caption="<img
border=0 src=\"images/arrow-right.gif\" alt=\"Next\">" />

        <!img alt="Next" src="images/arrow-right.gif"
border="0" WIDTH="13" HEIGHT="9">

         (...html omitted...)

      <table border=0>

      <s1ws:resultIteration formId="test" start="1" results="15">

         <tr class=body>

            <td valign=top>
```

```
                <s1ws:item property='number' />.  
                </td>
                <td>
                   <b><a href="<s1ws:item property='url'
/>"><s1ws:item property='title' /></a></b>
                   <br>
                   <s1ws:item property='passages' />
                   <font color="#999999" size="-2">
                   <s1ws:item property='url' /> -
                   <s1ws:item property='date' /> -
                   <s1ws:item property='size' /> KB
                   </font><br><br>
                </td>
             </tr>
          </s1ws:resultIteration>
          </table>
          (...html omitted...)
          <s1ws:resultNav formId="test" type="previous" />
          <s1ws:resultNav formId="test" type="full" offset="8" />
          <s1ws:resultNav formId="test" type="next" />
      (...html omitted...)
</s1ws:formSubmission>
```

The following figure shows the customized search results page:

Customized Search Results Page



The basic search result interface can be easily customized by manipulating the tags and modifying the HTMLs. For example, the navigation bar may be copied and placed before the search results. Users may also choose to show or not show any of the properties for a search record.

Besides being used along with a form, the `<search>`, `<resultIterate>` and related tags may be used to listed specific topics. The following sample code lists the top ten articles on Java Web Services on a site:

```
<s1ws:search Collection="Articles" Query="Java Web Services" />

<table cellspacing="0" cellpadding="3" border="0">

  <tr class="Title"><td>Java Web Services</td></tr>

</table>

<table cellspacing="0" cellpadding="3" border="0">

<s1ws:resultIteration>

<tr>

<td><a href="<s1ws:item property='URL' />"> <s1ws:item
property='Title'/></a></td>

</tr>

</s1ws:resultIteration>

</table>
```

Customizing Form and Results in Separate Pages

If you need the form and results pages to be separate, you must create the form page using the `<form>` set of tags and the results pages using the `<formAction>` set of tags.

A link to the form page needs to be added in the results page for a smooth flow of pages.

Tag Conventions

Note the following tag conventions:

- Classes for tags belong to the package `com.sun.web.search.taglibs`.

- All the `pageContext` attributes have the prefix `com.sun.web`. The attribute for search result for example, is `com.sun.web.searchresults.`*`form_id`* where *`form_id`* is the name of the form.
- Tag libraries are referenced with the prefix `s1ws`. Names of tags and their attributes are in mixed case with the first letter of each internal word capitalized, for example, `pageContext`.

Tag Specifications

Sun ONE Web Server includes a set of JSP tags that can be used to customize the search query and search results pages in the search interface.

For a complete list of JSP tags that you can use to customize your search pages, refer to the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications.*

Chapter 19

Web Publishing with WebDAV

Sun ONE Web Server 6.1 supports WebDAV or Web-based Distributed Authoring and Versioning, an emerging standard in Web-based collaboration. WebDAV is an extension to the HTTP/1.1 protocol that allows clients to perform remote web content authoring operations.

This chapter describes how you can use WebDAV on Sun ONE Web Server 6.1. It contains the following sections:

About WebDAV

WebDAV is an extension of the HTTP/1.1 protocol, and adds new HTTP methods and headers that provide authoring support for Web resources of any type, not only HTML and XML but also, text, graphics, spreadsheets, and all other formats.

Some of the tasks you can accomplish using WebDAV are:

- **Properties (meta-data) manipulation.** You can create, remove and query information about web pages, such as their authors and creation date using the WebDAV methods `PROPFIND` and `PROPPATCH`.
- **Collection and resource management.** You can create sets of documents and retrieve a hierarchical membership listing (similar to a directory listing in a file system) using the WebDAV methods `GET`, `PUT`, `DELETE`, and `MKCOL`.
- **Locking.** You can use WebDAV to keep more than one person from working on a document at the same time. The use of mutually exclusive or shared locks using the WebDAV methods `LOCK` and `UNLOCK`, helps to prevent the "lost updates" (overwriting of changes) problem.
- **Namespace operations.** You can use WebDAV to instruct the server to copy and move Web resources using the WebDAV methods `COPY` and `MOVE`.

WebDAV support in Sun ONE Web Server 6.1 provides the following features:

- Compliance with RFC2518 and interoperability with RFC2518 clients
- Security and access control for publishing
- Efficient publishing operations on file system-based WebDAV collections and resources

Common WebDAV Terminology

This section outlines the common terms you will encounter as you work with WebDAV.

URI. A URI (Uniform Resource Identifier) is a file identifier that provides an additional layer of security by using an abbreviated URL. The first part of the URL is substituted with a URL mapping that hides the file's full physical pathname from the user.

Source URI. The term, source URI, refers to the URI at which a resource's source can be accessed. To understand the concept of source URI, consider the following example:

A JSP page, `foo.jsp`, is located at the URI `/docs/date.jsp`. This page contains HTML markup and Java code which, when executed, prints today's date on the client's browser. When the server receives a GET request for `foo.jsp` from a client, before serving the page it executes the Java code. What the client receives is not `foo.jsp` as it resides on the server but instead a dynamically generated page that displays the current date.

If you were to create a source URI, say `/publish/docs`, and map it to the `/docs` directory containing `foo.jsp`, then a request for `/publish/docs/foo.jsp` would be a request for the source code of the `/docs/foo.jsp` JSP page. In this case, the server would serve the page without executing the Java code. The client would receive the unprocessed page exactly as stored on disk.

A request for the source URI is thus a request for the source of the resource.

Collection. A WebDAV collection is a resource or a set of resources that are enabled for WebDAV operations. A collection contains a set of URIs, termed member URIs, which identify member resources that are WebDAV-enabled.

Member URI. A URI which is a member of the set of URIs inside a collection.

Internal Member URI. A Member URI that is immediately relative to the URI of the collection. For example, if the resource with the URL `http://info.sun.com/resources/info` is WebDAV-enabled and if the resource with the URL `http://info.sun.com/resources/` is also WebDAV-enabled, then the resource with the URL `http://info.sun.com/resources/` is a collection and contains `http://info.sun.com/resources/info` as an internal member.

Property. A name/value pair that contains descriptive information about a resource. Properties are used for efficient discovery and management of resources. For example, a 'creationdate' property might allow for the indexing of all resources by the date on which the resources were created, and an 'author' property, for indexing by author name.

Live Property. A property that is enforced by the server. For example, the live `getcontentlength` property has as its value, the length of the entity returned by a GET request, which is automatically calculated by the server. Live properties include the following:

- The value of a property is read-only, maintained by the server
- The value of the property is maintained by the client, but the server performs syntax checking on submitted values.

Dead Property. A property that is not enforced by the server. The server only records the value of a dead property; the client is responsible for maintaining its consistency.

Sun ONE Web Server 6.1 supports the following live properties:

- `creationdate`
- `displayname`
- `getcontentlanguage`

- getcontentlength
- getcontenttype
- gettag
- getlastmodified
- lockdiscovery
- resourcetype
- supportedlock
- executable

NOTE Sun ONE web Server supports the live property executable that allows clients to change the file permissions associated with a resource.

An example of a PROPPATCH request for the executable live property:

```
PROPPATCH /test/index.html HTTP/1.1
Host: sun
Content-type: text/xml
Content-length: XXXX
<?xml version="1.0"?>
<A:propertyupdate xmlns:A="DAV:"
xmlns:B="http://apache.org/dav/props/">
<A:set>
<A:prop>
<B:executable>T</B:executable>
</A:prop>
</A:set>
</A:propertyupdate>
```

Locking. The ability to lock a resource provides a mechanism to guarantee that one user will not modify a resource while it is being edited by another. Locking prevents overwrite conflicts and resolves the "lost updates" problem.

Sun ONE Web Server supports two types of locking: shared and exclusive.

New HTTP Headers. WebDAV works by extending the HTTP/1.1 protocol. It defines new HTTP headers by which clients can communicate requests for WebDAV resources. These headers are:

- `Destination:`
- `Lock-Token:`
- `Timeout:`
- `DAV:`
- `If:`
- `Depth:`
- `Overwrite:`

New HTTP Methods. WebDAV introduces several new HTTP methods that instruct WebDAV-enabled servers how to handle requests. These methods are used in addition to existing HTTP methods such as `GET`, `PUT`, and `DELETE` to carry out WebDAV transactions. The new HTTP methods are briefly described below:

- `COPY`. Used to copy resources. Copying collections uses the `Depth:` header while the `Destination:` header specifies the target. The COPY method also uses the `Overwrite:` header, as appropriate.
- `MOVE`. Used to move resources. Moving collections uses the `Depth:` header while the `Destination:` header specifies the target. The MOVE method also uses the `Overwrite:` header, as appropriate.
- `MKCOL`. Used to create a new collection. This method is used to avoid overloading the PUT method.
- `PROPPATCH`. Used to set, change, or delete properties on a single resource.
- `PROPFIND`. Used to fetch one or more properties belonging to one or more resources. When a client submits a PROPFIND request on a collection to the server, the request may include a Depth: header with a value of `0`, `1`, or `infinity`.
    - `0`. Specifies that the properties of the collection at the specified URI will be fetched.
    - `1`. Specifies that the properties of the collection and resources immediately under the specified URI will be fetched.
    - `infinity`. Specifies that the properties of the collection and all member URIs it contains will be fetched. Be aware that because a request with infinite depth would crawl the entire collection, it could impose a large burden on the server.

- LOCK. Adds locks on resources. Uses the `Lock-Token:` header.
- UNLOCK. Removes locks from resources. Uses the `Lock-Token:` header.

Using WebDAV

A complete WebDAV transaction involves a WebDAV-enabled server, such as Sun ONE Web Server 6.1, that can service requests for WebDAV resources, as well as a WebDAV-enabled client such as Adobe® GoLive® or Macromedia® DreamWeaver® that supports WebDAV-enabled Web publishing requests.

On the server-side, you need to enable and configure Sun ONE Web Server 6.1 to be able to service WebDAV requests.

To configure Sun ONE Web Server 6.1 to use WebDAV, the following steps are needed:

- Enabling WebDAV
- Creating a WebDAV Collection
- Configuring WebDAV
- Enabling Access Control for WebDAV

Enabling WebDAV

When you install Sun ONE Web Server 6.1, WebDAV is disabled by default.

In order to enable WebDAV at the collection level, you need to also enable WebDAV at the server instance level and the virtual server class level.

NOTE The attributes specified on a collection override attribute values set at the virtual server level.

The different levels at which you can enable WebDAV are described in the following sections:

- Enabling WebDAV for the Server Instance
- Enabling WebDAV for a Virtual Server Class

Enabling WebDAV for the Server Instance

You can use the Administration Server to enable WebDAV for the entire server. When you do so, the following directive is added to the `magnus.conf file` that loads the WebDAV plugin:

```
Init fn="load-modules" shlib="/s1ws6.1/lib/libdavplugin.so"
funcs="init-dav,ntrans-dav,pcheck-dav,service-dav"

shlib_flags="(global|now)"

Init fn="init-dav" LateInit=yes
```

The `init-dav` Init function initializes and registers the WebDAV subsystem.

To enable WebDAV globally, perform the following tasks:

1. Access the Server Manager of the server you want to enable WebDAV for.
2. Click the Enable / Disable WebDAV link on the Preferences tab.
3. Check the Enable WebDAV Globally checkbox.

Enabling WebDAV for the Instance

4. Click Apply.
5. Click the Apply Changes button to restart the server

   or

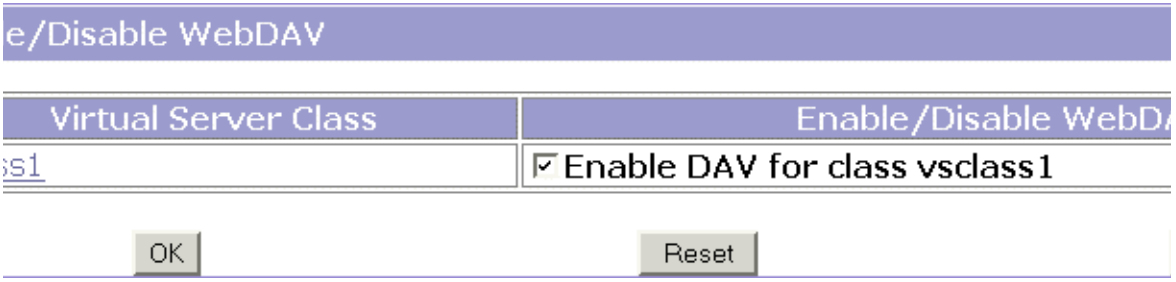
   click Load Configuration Files to dynamically apply your changes.

Enabling WebDAV for a Virtual Server Class

To enable WebDAV for a particular virtual server class:

1. Select the virtual server class.
2. Click the Content Mgmt tab.
3. Click the Enable / Disable WebDAV link.

   Enabling WebDAV for a virtual server class.



4. Check the Enable DAV checkbox.
5. Click OK.

When you enable WebDAV for a virtual server class, the associated `obj.conf` file is updated with the following entries:

```
<Object name="default">
...
Service fn="service-dav"
method="(OPTIONS|PUT|DELETE|COPY|MOVE|PROPFIND|PROPPATCH|LOCK|UN
LOCK|MKCOL)"
Error fn="error-j2ee"
...
</Object>
...
<Object name="dav">
PathCheck fn="check-acl" acl="dav-src"
Service fn="service-dav"
method="(GET|HEAD|POST|PUT|DELETE|COPY|MOVE|PROPFIND|PROPPATCH|L
OCK|UNLOCK|MKCOL)"
</Object>
```

Enabling WebDAV for a Collection

If you have added one or more WebDAV collections to a virtual server, you can choose to disable and enable them at any time. For information on how you can do so, see “Editing a WebDAV Collection” on page 427.

Creating a WebDAV Collection

A WebDAV collection is a resource or a set of resources that are enabled for WebDAV operations. These operations include web publishing and collaborative authoring, namespace management, and metadata management.

To add a WebDAV collection to a virtual server, perform the following tasks:

1. Make sure that WebDAV is enabled for the server instance and for the virtual server class. For more information, see “Enabling WebDAV for the Server Instance” on page 423 and “Enabling WebDAV for a Virtual Server Class” on page 424.
2. Access the virtual server you want to manage and click the WebDAV tab.
3. In the Add DAV Collection page, enter the following information:
    - **URI** (required). The URI by which the content is accessed.

- **Source URI** (optional). The URI by which the source is accessed.

NOTE If you plan to publish dynamic content such as CGI or SHTML, you must have a source URI configured.

For an explanation of the term source URI, see Common WebDAV Terminology.

- **Lock Database** (optional). The directory where the locking database will be maintained. The default value is *server-instance*/lock-db/vs-id.
- **Minimum Lock Timeout** (optional). The minimum lifetime of a lock in seconds. The default value is 0. For more information, see Minimum Lock Timeout.
- **Limit XML Request Body** (optional). The maximum size of the XML content in the body of the request. Restrict the size to prevent the possibility of Denial of Service (DOS) attacks.
- **Maximum Property Depth** (optional). The depth of the PROPFIND request.
  - 0 applies only to the specified resource.
  - 1 applies to the specified resource and the next level of resources it contains.
  - infinity applies to the specified resource and all the resources it contains.

  By default, the value is set to 0.
- **Enabled** (optional). Enables WebDAV functionality for the collection.

4. Click OK.

NOTE
- When you use the Administration server to add a collection, the server does not automatically create a directory for the collection on the filesystem. It is the responsibility of the administrator to ensure that a directory corresponding to the collection is created on the filesystem.
- On UNIX systems, if you have installed the Web Server as root (superuser), and are running the server as a different user, ensure that the user you are running the server as has read and write permissions on the directories corresponding to the WebDAV collections you create.

Editing a WebDAV Collection

You can edit the attributes of an existing DAV collection, for example to configure access control on the collection.

To edit an existing WebDAV collection, perform the following tasks:

1. Access the virtual server on which the collection exists and click the WebDAV tab.
2. In the Edit DAV Collections page, modify the following information:
   - **Delete**. Allows you to or delete a collection.
   - **URI**. Displays the URI by which the content is accessed.
   - **Enabled**. Indicates whether WebDAV is enabled (`true`) or disabled (`false`).
   - **Edit Collection**. Click this button to configure the following:
     - **URI** (required). The URI by which the content is accessed.
     - **Source URI** (optional). The URI by which the source is accessed.
     - **Lock Database** (optional). The directory where the locking database will be maintained.
     - **Minimum Lock Timeout** (optional). The minimum lifetime of a lock in seconds. For more information, see Minimum Lock Timeout.

NOTE If the value of `minlocktimeout` is -1, it indicates an infinite lock.

     - **Limit XML Request Body** (optional). The maximum size of the XML content in the body of the request.
     - **Maximum Property Depth** (optional). The depth of the `PROPFIND` request.

       - `0` applies only to the specified resource.

       - `1` applies to the specified resource and the next level of resources it contains.

       - `infinity` applies to the specified resource and all the resources it contains.

       By default, the value is set to 0.

- **Enabled** (optional). Enables WebDAV functionality for the collection.

- **Edit ACL.** Click this to set up access control restrictions for this collection or URI.

Configuring WebDAV

You might want to configure WebDAV for several reasons: for example, to tune server performance, to eliminate security risks, or to provide for conflict-free remote authoring.

To suit your configuration requirements, you can change the minimum amount of time the server holds a lock on a WebDAV resource, the depth of the PROPFIND request on a collection, and the maximum size of the XML content allowed in the body of a request, and so on.

Default WebDAV attributes can be configured at the virtual server level for all collections under a virtual server. The values configured here correspond to the `DAV` element in the `server.xml` file.

WebDAV attributes can also be configured at a collection level and override any virtual-server -level attributes configured for the collection. The attribute values configured at the collection level correspond to the `DAVCOLLECTION` element in the `server.xml` file.

- Configuring WebDAV at the Virtual Server Level
- Configuring WebDAV at the URI Level

Configuring WebDAV at the Virtual Server Level

To configure WebDAV functionality for the virtual server, you need to edit the attributes of the `DAV` object. You can do this either by using the Administration Server or by manually editing the `server.xml` file.

The following table describes the attributes of the `DAV` object you can configure:

Table 1 Attributes of the DAV object

Attribute	Description
`enabled`	Specifies if WebDAV functionality is enabled for this virtual server. This is an optional attribute. The default value is `true`. Possible values are `true` and `false`.
`lockdb`	Specifies the directory where the locking database will be maintained. This is an optional attribute.
`minlocktimeout`	Specifies the minimum lifetime of a lock in seconds. This value indicates the amount of time that an element will be locked before the lock is automatically removed. For more information, see Minimum Lock Timeout. This is an optional attribute.
`maxxmlrequestbodysize`	Specifies the maximum size of the XML content in the body of the request. This is an optional attribute. The default value is 8K. Restrict the size to prevent the possibility of Denial of service (DOS) attacks.
`maxpropdepth`	Specifies the depth of the PROPFIND request. This is an optional parameter. The default value is `0`. Prevent excessive memory consumption by restricting the size of this parameter.

Configuring WebDAV at the URI Level

To configure WebDAV functionality at the URI level, you need to edit the attributes of the `DAVCOLLECTION` object in the `server.xml` file.

The following table describes the attributes of the `DAVCOLLECTION` object that you can configure:

Table 2 Attributes of the `DAVCOLLECTION` object

Attribute	Description
`enabled`	Specifies if DAV functionality is enabled for this collection. This is an optional attribute. Possible values are `true` and `false`. The default value is `true`.
`uri`	Specifies the URI by which the content is accessed. This is a required attribute.
`sourceuri`	Specifies the URI by which the source is accessed. For more information, see Common WebDAV Terminology and Using Source URI and Translate:f Header on a WebDAV-Enabled Server. This is an optional attribute. If the `sourceuri` is not specified, the default behavior is to deny access to the source of any dynamic content in the collection. You can specify the same URI for both `uri` and `sourceuri`, in which case the server will always returns the source of dynamic content. This may be useful if you use a separate, secured virtual server for publishing.
`lockdb`	Specifies the directory where the locking database will be maintained. This is an optional attribute.
`minlocktimeout`	Specifies the minimum lifetime of a lock in seconds. This value indicates the amount of time that an element will be locked before the lock is automatically removed. For more information, see Minimum Lock Timeout. This is an optional attribute.
`maxxmlrequestbodysize`	Specifies the maximum size of the XML content in the body of the request. This is an optional attribute. Restrict the size to prevent the possibility of Denial of Service (DOS) attacks.

Table 2 Attributes of the `DAVCOLLECTION` object

Attribute	Description
`maxpropdepth`	Specifies the depth of the PROPFIND request, which lists the member resources of a collection. This is an optional parameter. Prevent excessive memory consumption by restricting the size of this parameter.

Using Source URI and Translate:f Header on a WebDAV-Enabled Server

WebDAV methods operate on the source of a resource or a collection. HTTP methods such as GET and PUT are overloaded by the WebDAV protocol and therefore, a request with these methods can either be a request to the source of the resource or a request to the content (output) of the resource.

Microsoft and many other WebDAV vendors have addressed this problem by sending a `Translate:f` header with the request to inform the server that the request is for the source. In order to be interoperable with the popular WebDAV client Microsoft WebFolders, Sun ONE Web Server 6.1 recognizes the `Translate:f` header as a request to the source of the resource. To accommodate clients that do not send the `Translate:f` header, Sun ONE Web Server 6.1 defines a source URI. See Common WebDAV Terminology for a more detailed explanation of the term source URI.

For a WebDAV-enabled collection, the request to the URI retrieves the content (output) of the resource and a request to the source URI retrieves the source of the resource. A request to the URI with a `Translate:f` header is treated as a request to the source URI.

Note that by default all access to the source of a resource is denied by the `dav-src` ACL with the following declaration in the server instance-specific ACL file:

```
deny (all) user = "anyone";
```

An user can enable access to the source to a user by adding access rights to the source URI. For more information on adding URI-specific ACLs, please see Enabling Access Control for WebDAV.

Locking and Unlocking Resources

Sun ONE Web Server allows the server administrator to lock a resource so as to serialize access to that resource. Using a lock, a user accessing a particular resource is reassured that another user will not modify the same resource. In this way, the "lost updates" problem is resolved as multiple users share resources on the server. The lock database maintained by the server keeps track of the lock tokens issued and in use by clients.

Sun ONE Web Server 6.1 supports the `opaquelocktoken` URI scheme, which is designed to be unique across all resources for all time. This uses the Universal Unique Identifier (UUID) mechanism, as described in ISO-11578.

Sun ONE Web Server 6.1 recognizes two types of locking mechanisms:

- Exclusive Locks
- Shared Locks

Exclusive Locks

An exclusive lock is a lock that grants access to a resource to only a single user. Another user can access the same resource only after the exclusive lock on the resource is removed.

Exclusive locking sometimes proves to be too rigid and expensive a mechanism for locking resources. For example, in the event of a program crash or the lock owner forgetting to unlock the resource, a lock timeout or the administrator's intervention would be required to remove the exclusive lock.

Shared Locks

A shared lock allows multiple users to receive a lock to a resource. Hence any user with appropriate access can get the lock.

When using shared locks, lock owners may use any other communication channel to coordinate their work. The intent of a shared lock is to let collaborators know who else may be working on a resource.

Lock Management

Sun ONE Web Server 6.1 provides a lock management feature which allows you to view all outstanding locks, their type, the resources they hold, the duration of the lock and so on.

To use lock management, do the following:

1. Access the WebDAV-enabled virtual server.
2. Click the WebDAV tab.
3. Click the Lock Management link.
4. Select the locking database and the WebDAV-enabled URI for which you want to view the outstanding locks and other information.
5. Click List Lock Info.

Minimum Lock Timeout

You can control locking by configuring the value of the `minlocktimeout` attribute of the `DAV` or `DAVCOLLECTION` objects in the `server.xml` file. The `minlocktimeout` attribute specifies the minimum lifetime of a lock in seconds. This value indicates the amount of time that an element will be locked before the lock is automatically removed.

This is an optional attribute. If the value is set to -1, the lock will never expire. Setting the value to `0` allows all the resources in the collection to be locked with the Timeout header specified in the request.

If no Timeout header is specified, then the resource is locked with infinite timeout. If a request has a Timeout header set to the value `Infinite`, then also, the resource is locked with infinite timeout.

If the request for a WebDAV resource has a Timeout header value that is equal to or greater than the `minlocktimeout` value specified in `server.xml`, then the resource is locked for the period of time specified in the request.

However, if the request has a Timeout header value that is lower than the `minlocktimeout` value specified in `server.xml`, then the resource is locked with the `minlocktimeout` value specified in `server.xml`.

The following table illustrates how Sun ONE Web Server handles locking requests:

Table 3 How Sun ONE Web Server handles locking requests

If Timeout header value in Request is set to:	The resource is:
`Infinite`	Locked with timeout set to -1 (infinite)
None	Locked with timeout set to -1 (infinite)
`Second-`*xxx*	• Locked with *xxx* value, if *xxx* is equal to or greater than `minlocktimeout` value set in `server.xml` or • locked with `minlocktimeout` value specified in `server.xml`, if *xxx* is lower than `minlocktimeout` value set in `server.xml`.

Example of a Lock Request

This example illustrates a request for an exclusive write lock on the resource `/col1/myfile.html` with a timeout of 500 seconds.

```
LOCK /col1/myfile.html HTTP/1.1
Host: sun
Content-Type: text/xml; charset="utf-8"
Content-Length: 259
Timeout: Second-500
<?xml version="1.0" encoding="utf-8" ?>
<d:lockinfo xmlns:d="DAV:">
    <d:locktype><d:write/></d:locktype>
    <d:lockscope><d:exclusive/></d:lockscope>
    <d:owner>
    <d:href>http://info.sun.com/resources/info.html</d:href>
    </d:owner>
</d:lockinfo>
```

Enabling Access Control for WebDAV

You can control who accesses WebDAV-enabled documents and directories and what operations different users or different groups of users can perform upon the files. You can also completely prohibit access to a file or folder or you can restrict access to certain authenticated users.

If the default access control (ACL) that governs your server is not restrictive or flexible enough for your needs, you can use the Restrict Access function (choose Server Preferences and click the Restrict Access link) to create an ACL that is more appropriate for restricting access to WebDAV-enabled resources.

WebDAV requests are authenticated and authorized by the AuthTrans and PathCheck NSAPI stages respectively. In the following example, an access control rule is defined that denies write and delete access to the collection `/catalog` to all except a user named "joe":

```
acl "uri=/catalog/*";

deny(all)

user="anyone";

allow (read,list,execute,info)

user = "all";

allow(write,delete)

user="joe";
```

For more details, refer to Editing a WebDAV Collection.

Restricting Access on WebDAV-Enabled Resources

Access control for a WebDAV collection is specified using native ACL files. Every WebDAV method requires a particular access right to a WebDAV-enabled resource. For example, if a WebDAV-enabled file is to be shared by concurrent users, in order to lock or unlock the resource for concurrency control, a write permission to the resource is required.

The following table below summarizes the rights required for WebDAV methods.

Table 19-4 Rights required for WebDAV

DAV Methods	Access Rights Needed
DELETE	delete
PROPFIND	read

Table 19-4 Rights required for WebDAV

DAV Methods	Access Rights Needed
PROPPATCH	write
LOCK / UNLOCK	write
MKCOL	write
COPY(*src*,*dst*)	*src* - read
	dst - write
MOVE(*src*,*dst*)	*src* - delete
	dst - write
GET on request-uri	read
GET on request-uri	read
Translate:f	
PUT on request-uri	write
PUT on request-uri	write
Translate:f	

Security Considerations

When you use WebDAV, keep in mind the following security considerations:

- Ensure that a WebDAV-enabled server process has read / write permissions to the file systems that need to be controlled.
- For security reasons, you may wish to configure WebDAV-enabled virtual servers on a different listen socket, one that has restricted access and uses SSL to encrypt transmitted data. See Using Certificates and Keys for more information on using SSL.
- Prevent Denial of Service (DOS) attacks by restricting the size of the XML content in the request body. By default, the size is restricted to 8K.
- Because Basic authentication uses cleartext to transmit authentication details, unless your connection is secure, use Digest rather than Basic authentication to authenticate WebDAV clients.
- Because PROPFIND requests run the potential risk of unwanted access to server contents, use access control techniques to secure WebDAV-enabled resources.

- WebDAV, through its source URI facility, can potentially expose URIs containing sensitive information such as script resources. You should be aware of the risks of allowing the remote authoring of scripts, and should limit read and write access to source resources to authorized users only.
- Prevent excessive memory consumption by restricting the depth of PROPFIND requests. By default, the depth is restricted to 0.

Part 5

Appendixes

Appendix A

Command Line Utilities

This appendix contains instructions for working with the `HttpServerAdmin` command line utility.

HttpServerAdmin (Virtual Server Administration)

`HttpServerAdmin` is a command line utility that performs the same administrative functions as the virtual server user interface in the Server Manager and the Class Manager. If you prefer to set up your virtual servers using the command line interface, use `HttpServerAdmin`.

NOTE To use the `HttpServerAdmin` command line utility, you must have superuser privileges on the system.

The `HttpServerAdmin` command line utility is located in the *`server_root`*`/bin/https/httpadmin/bin` directory.

Before you can run `HttpServerAdmin`, you need to set the environment variable IWS_SERVER_HOME to the server root directory in your environment.

For example, on UNIX/Linux systems:

```
setenv IWS_SERVER_HOME /usr/sun/servers
```

On Windows systems:

1. On the Control Panel, choose System.
2. Click the Environment tab.
3. Type IWS_SERVER_HOME in the Variable field and the path to your server root in the Value field.

4. Click Set.
5. Click OK.

NOTE In order to perform all commands, you need to have write permissions to the file `server.xml` where the virtual server information is stored.

HttpServerAdmin Syntax

The `HttpServerAdmin` syntax is as follows:

`HttpServerAdmin` *command_name command_options* `-d` *server_root* `-sinst` *http_instance*

You can get an online explanation of the command parameters by typing the following command:

```
./HttpServerAdmin -h
```

There are four possible values for the *command_name* parameter:

- control
- create
- delete
- list

Each command has its own set of command options. For more information, see the sections in this chapter that describe each command.

Regardless of the value of the command parameter, the parameters shown in Table A-1 can apply to all uses of the `HttpServerAdmin` command.

Table A-1 HttpServerAdmin Parameters

Parameter	Value
`-d` *server_root*	(required). This parameter designates the path to the server root (the location where the server is installed).
`-sinst` *http_instance*	(required). This parameter designates which instance `HttpServerAdmin` affects.

control Command

Use the `control` command to start, stop, and disable classes and virtual servers. If you do not specify a virtual server, the command starts, stops or disables every virtual server in the class.

Options

Use the options shown in Table A-2 with the `control` command to control classes and virtual servers.

Table A-2 Control command options

Options	Value
-start	Starts the specified virtual server, or all virtual servers in the class if no virtual server is specified.
-stop	Stops the specified virtual server, or all virtual servers in the class if no virtual server is specified.
-disable	Disables the specified virtual server, or all virtual servers in the class if no virtual server is specified.

Syntax

`HttpServerAdmin control -cl` *classname*`, -`*control_option* `[-id` *virtual_server*`] -d` *server_root* `-sinst` *http_instance*

Parameters

Use these parameters with the command options to control virtual servers

Table A-3 Control command parameters

Parameters	Value
-cl *classname*	Designates the virtual server class
-id *virtual_server*	(optional) Designates the virtual server ID you are controlling.

Examples

```
HttpServerAdmin control -cl myclass -start -id myvirtualserver -d
/usr/sun/servers -sinst https-sun.com

HttpServerAdmin control -cl myclass -stop -id myvirtualserver -d
/usr/sun/servers -sinst https-sun.com

HttpServerAdmin control -cl myclass -disable -id myvirtualserver
-d /usr/sun/servers -sinst https-sun.com
```

create Command

Use the `create` command to create classes of virtual servers, virtual servers, and listen sockets.

Options

Use the options shown in Table A-4 with the `create` command to create classes, listen sockets, virtual servers, and resources.

Table A-4 Create command options

Option	Value
-c	Creates a virtual server class.
-l	Creates a listen socket.
-v	Creates a virtual server.
-r	Creates a resource.

Each of these options in turn has its own parameters, which are shown in the following sections.

Create Virtual Server Class

Use this option of the create command to create a virtual server class.

Syntax

`HttpServerAdmin create -c -cl` *classname* `-docroot` *document_root* `[-obj` *obj.conf_file*`] [-acptlang` *accept_language*`] -d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-5 with the `create -c` command option to create classes.

Table A-5 Create virtual server class parameter

Parameter	Value
-cl *classname*	The name of the class you want to create.
-docroot *document_root*	The document root for the class. This has to be an absolute path.
-obj *obj.conf_file*	(optional) The obj.conf file for the class. If you do not specify this parameter, the server creates the obj.conf file as *classname*.obj.conf. If you want a different name for the class' obj.conf file, specify it here.
-acptlang *accept_language*	(optional) If you do not specify this parameter, acptlang will be off by default.

Example

```
HttpServerAdmin create -c -cl myclass1 -docroot /docs -d
/export/sun/servers -sinst https-sun.com
```

Create Listen Socket

Use this option of the create command to create a listen socket.

Syntax

`HttpServerAdmin create -l -ip` *ip_address* `-port` *port_number* `-sname` *server_name* `-id` *default_virtual_server* `[-sec security] [-acct` *number_of_accept_threads*`] -d` *server_root* `-sinst http_instance`

Parameters

Use the parameters shown in Table A-6 with the `create -l` command option to create listen sockets.

Table A-6 Create listen socket parameters

Parameter	Value
-ip *ip_address*	The IP address for the listen socket.
-port *port_number*	The port number for the listen socket.
-sname *server_name*	The server name to associate with the listen socket.
-id *default_virtual_server*	The ID of the default virtual server. This virtual server must exist before you can use it to create a listen socket.
-acct *number_of_accept_threads*	(optional) The number of accept threads for the listen socket.
-sec *on*	(optional) If specified, use on to enable security for the listen socket. If not specified, security is not enabled.

Example

```
HttpServerAdmin create -l -id ls3 -ip 0.0.0.0 -port 1333 -sname
austen -defaultvs vs2 -sec on -acct 4 -d /export/carey/server6
-sinst https-austen.com
```

Create Virtual Server

Use this option of the create command to create a virtual server.

Please note that if you do not include values for some of the optional parameters, defaults are provided. You can always change the default values after the virtual server is created.

Syntax

`HttpServerAdmin create -v -id` *virtual_server* `-cl` *classname* `-urlh` *urlhosts* `[-state` *state*`][-docroot` *document_root*`] [-mime` *mime_types_file*`] [-aclid` *acl_ID*`] -d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-7 with the `create -v` command option to create virtual servers.

Table A-7 Create listen socket parameters

Parameter	Value
-id *virtual_server*	The ID of the virtual server you are creating.
-cl *classname*	The class of which the virtual server will be a member.
-urlh *URL_hosts*	The URL hosts for the virtual server. You can specify more than one URL host, separated by a comma.
-state *state*	(optional) Valid values are on, off, and disable.
-docroot *document_root*	(optional) If you want to specify a document root for a virtual server, use this parameter. You must use an absolute path name.
-mime *mime_types_file*	(optional) The name of the MIME types file for the virtual server.
-aclid *acl_ID*	(optional) The ACL file ID <ACLID> used in the `server.xml` file

Examples

```
HttpServerAdmin create -v -id vs3 -cl class1 -urlh annh -d
/export/sun/server6 -sinst https-sun.com

HttpServerAdmin create -v -id vs4 -cl class1 -urlh annh,annh2
-state off -mime mime.types -d /export/sun/server6 -sinst
https-sun.com
```

Create JDBC Connection Pool

Use the `create -r` command to create a new JDBC connection pool using the Command Line Interface.

Syntax

`HttpServerAdmin -create -r -jdbcconnectionpool -poolname` *jdbcpoolname* `-classname` *classname* [`-steadypoolsize` *steadypoolsize*] [`-maxpoolsize` *maxpoolsize*] [`-poolresizequantity` *poolresizequantity*] [`-idletimeout` *idletimeout*] [`-maxwaittime` *maxwaittime*] [`-connectionvalidation` *true*/*false*] [`-connectionvalidationmethod` *connectionvalidationmethod*] [`-validationtablename` *validationtablename*] [`-failall` *true*/*false*] [`-desc` *description*] [[`-property` *propertyname*=*value*],...]

Options

The following table summarizes all the options that you need to create connection pools with the `create -r` command option.

Table A-8 Create connection pool parameters

Parameter	Value
`poolname` *`jdbcpoolname`*	The pool name for the JDBC connection pool.
`classname` *`classname`*	The vendor-specific classname that implements the data source.
`steadypoolsize` *`steadypoolsize`*	The minimum number of connections that must be maintained in the pool
`maxpoolsize` *`maxpoolsize`*	The maximum number of connections allowed in the pool.
`poolresizequantity` *`poolresizequantity`*	The size of the batch by which the pool is resized when the `steadypoolsize` value is approached.
`idletimeout` *`idletimeout`*	The maximum time in seconds that a connection can remain idle in the pool.
`maxwaittime` *`maxwaittime`*	The amount of time the caller will wait before getting a connection timeout.
`connectionvalidation` *`true/false`*	Specifies whether connections will be validated before they are passed to the application.
`connectionvalidationmethod` *`connectionvalidationmethod`*	The methods that can employ to validate database connections. Legal values are `auto-commit`, `meta-data`, and `table`.

Table A-8 Create connection pool parameters

Parameter	Value
`validationtablename` *`validationtablename`*	The name of the table if `connectionvalidationmethod` is set to `table`.
`failall` *`true/false`*	Specifies whether to fail all connections in the pool and re-establish them if a single connection is determined to have failed.
`desc` *`description`*	The description of the pool.
`property` *`propertyname=value`*	The name-value pairs that specify standard and proprietary JDBC connection pool properties

Example

```
HttpServerAdmin create -r -jdbcconnectionpool -poolname testpool
-classname "oracle.jdbc.pool.OracleDataSource" -property
"URL=jdbc:oracle:thin:@dbhost:1521:ORCL,user=scott,password=tige
r" -d /opt/Sun/S1WS6.1 -sinst testinstance
```

Create JDBC Resource

Use the `create -r` command to create a new JDBC resource using the Command Line Interface.

Syntax

`HttpServerAdmin -create -r -jdbc -jndiname` *jndiname* `-poolname` *poolname* [`-desc` *description*] [`-enabled` *true/false*]

Options

The following table summarizes all the options that you need to create a new JDBC resource with the `create -r` command option.

Table A-9 Create JDBC resource parameters

Parameter	Value
`jndiname` *`jndiname`*	The JNDI name of the resource.
`poolname` *`poolname`*	The pool name for the JDBC connection pool.
`desc` *`description`*	The description of the pool.
`enabled` *`true/false`*	Specifies whether the resource is enabled or disabled. If a JDBC resource is disabled, no application component can connect to it, but its configuration remains in the server instance.

Example

```
HttpServerAdmin create -r -jdbc -jndiname "jdbc/testjdbcresource"
-poolname testpool -d /opt/Sun/S1WS6.1 -sinst testinstance
```

Create Custom Resource

Use the `create -r` command to create a new custom resource using the Command Line Interface.

Syntax

`HttpServerAdmin -create -r -custom -jndiname` *jndiname* `-resourcetype` *resourcetype* -factoryclass *factoryclassname* [`-enabled` *true/false*] [`-desc` *description*] [[`-property` *propertyname=value*],...]

Options

The following table summarizes all the options that you need to create a new JDBC resource with the `create -r` command option.

Table A-10 Create custom resource parameters

Parameter	Value
`jndiname` *`jndiname`*	The JNDI name of the resource.

Table A-10 Create custom resource parameters

Parameter	Value
`resourcetype` *`resourcetype`*	The resource type.
`factoryclassname` *`factoryclassname`*	The classname of the object factory.
`enabled` *`true/false`*	Specifies whether the resource is enabled or disabled.
`desc` *`description`*	The description of the pool.
`property` *`propertyname=value`*	The name-value pairs that specify the properties of the custom resource.

Example

```
HttpServerAdmin create -r -custom -jndiname "testcustomresource"
-resourcetype "java.lang.String" -factoryclass
"com.mycom.test.StringFactory" -d /opt/Sun/S1WS6.1 -sinst
testinstance
```

Create External JNDI Resource

Use the `create -r` command to create a new external JNDI resource using the Command Line Interface.

Syntax

`HttpServerAdmin -create -r -external -jndiname` *jndiname* `-jndilookupname` *jndilookupname* `-restype` *restype* -factoryclass *factoryclass* [`-enabled` *true/false*] [`-desc` *description*] [[`-property` *propertyname=value*],...]

Options

The following table summarizes all the options that you need to create a new external JNDI resource with the `create -r` command option.

Table A-11 Create external JNDI resource parameters

Parameter	Value
`jndiname` *`jndiname`*	The JNDI name of the resource.

Table A-11 Create external JNDI resource parameters

Parameter	Value
`jndilookupname` *`jndilookupname`*	The JNDI lookup name for the resource.
`restype` *`restype`*	The resource type.
`factoryclass` *`factoryclass`*	The classname of the object factory.
`enabled` *`true/false`*	Specifies whether the resource is enabled or disabled.
`desc` *`description`*	The description of the pool.
`property` *`propertyname=value`*	The name-value pairs that specify the properties of the custom resource.

Example

```
HttpServerAdmin create -r -external -jndiname
"testexternalresource" -jndilookupname "rmiconverter" -restype
"samples.rmi.simple.ejb.ConverterHome" -factoryclass
"com.sun.jndi.cosnaming.CNCtxFactory" -property
"java.naming.provider.url=iiop://localhost:3700" -d
/opt/Sun/S1WS6.1 -sinst testinstance
```

Create Mail Resource

Use the `create -r` command to create a new mail resource using the Command Line Interface.

Syntax

`HttpServerAdmin -create -r -mail -jndiname` *jndiname* `-host` *host* `-user` *user* `-from` *from* `[-storeprotocol` *storeprotocol*`] [-storeprotocolclass` *storeprotocolclass*`] [-transportprotocol` *transportprotocol*`] [-transportprotocolclass` *transportprotocolclass*`] [-enabled` *true/false*`] [-desc` *description*`] [[-property` *propertyname*=*value*`]...]`

Options

The following table summarizes all the options that you need to create a new mail resource with the `create -r` command option.

Table A-12 Create mail resource parameters

Parameter	Value
`jndiname` *`jndiname`*	The JNDI name of the resource.
`host` *`host`*	The mail server host name.
`user` *`user`*	The mail server user name.
`from` *`from`*	The e-mail address the mail server uses to indicate the message sender.
`storeprotocol` *`storeprotocol`*	Specifies the storage protocol service, which connects to a mail server, retrieves messages, and saves messages in folder(s). Example values are `imap` and `pop3`.
`storeprotocolclass` *`storeprotocolclass`*	Specifies the service provider implementation class for storage. You can find this class at: • `http://java.sun.com/products/javamail/` • `http://java.sun.com/products/javabeans/glassgow/jaf.html`
`transportprotocol` *`transportprotocol`*	Specifies the transport protocol service, which sends messages.
`transportprotocolclass` *`transportprotocolclass`*	Specifies the service provider implementation class for transport. You can find this class at: • `http://java.sun.com/products/javamail/` • `http://java.sun.com/products/javabeans/glassgow/jaf.html`
`enabled` *`true/false`*	Determines whether this resource is enabled at runtime. Legal values are `on`, `off`, `yes`, `no`, `1`, `0`, `true`, `false`.
`desc` *`description`*	A description of the resource.
`property` *`propertyname=value`*	The name-value pairs that specify the properties of the custom resource.

Example

```
HttpServerAdmin create -r -mail -jndiname "localmail" -host
localhost -user mailid -from mailid@mailhost -d /opt/Sun/S1WS6.1
-sinst testinstance
```

delete Command

Use the delete command to delete classes of virtual servers, virtual servers, and listen sockets.

Options

Use the options shown in Table A-13 with the `delete` command to delete classes.

Table A-13 Delete command options

Option	Value
-c	Deletes the specified virtual server class.
-l	Deletes the specified listen socket IDs
-v	Deletes the specified virtual servers.
-r	Deletes the specified resource.

Delete Class

Use this option of the delete command to delete a virtual server class.

Syntax

`HttpServerAdmin delete -c -cl` *classname* `-d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-13 with the `delete` command to delete classes.

Table A-14 Delete class parameters

parameter	Value
-c l *class*	The class name you want to delete.

Example

```
HttpServerAdmin delete -c -cl class1 -d /export/sun/server6
-sinst https-sun.com
```

Delete Listen Socket

Use this option of the delete command to delete a listen socket.

Syntax

`HttpServerAdmin delete -l -id` *listen_socket* `-d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-13 with the `delete` command to delete classes.

Table A-15 Delete class parameters

parameter	Value
-id *listen_socket*	The ID of the listen socket you want to delete.

Example

```
HttpServerAdmin delete -l -id ls3 -d /export/sun/server6 -sinst
https-sun.com
```

Delete Virtual Server

Use this option of the delete command to delete a virtual server.

Syntax

`HttpServerAdmin delete -v -id` *virtual_server* `-cl` *classname* `-d` *server_root* `-sinst` *http_instance*

Parameters

Use the parameters shown in Table A-13 with the `delete` command to delete a virtual server.

Table A-16 Delete virtual server parameters

parameter	Value
-id *virtual_server*	The virtual server ID you want to delete
-cl *class*	The class the virtual server belongs to.

Example

```
HttpServerAdmin delete -v -id vs3 -cl class1 -d
/export/sun/server6 -sinst https-sun.com
```

Delete JDBC Connection Pool

Use this option of the delete command to delete a connection pool.

Syntax

`HttpServerAdmin delete -r` *jdbcconnectionpoolname*

Parameters

Use the parameters shown in Table A-13 with the `delete` command to delete a connection pool.

Table A-17 Delete connection pool parameters

parameter	Value
connectionpoolname	The name of the connection pool you want to delete

Example

```
HttpServerAdmin delete -r connpool
```

Delete JNDI Resource

Use this option of the delete command to delete a JNDI resource.

Syntax

`HttpServerAdmin delete -r` *jndiname*

Parameters

Use the parameters shown in Table A-13 with the `delete` command to delete a JNDI resource.

Table A-18 Delete JNDI resource parameters

parameter	Value
jndiname	The JNDI name of the resource you want to delete

Example

```
HttpServerAdmin delete -r testresource
```

list Command

Use the `list` command to list classes of virtual servers, virtual servers, listen sockets, and resources.

Syntax

`HttpServerAdmin list -`*command_option* `-d` *server_root* `-sinst` *http_instance*

Options

Table A-19 List command options

Option	Value
-c	Lists all virtual server classes.
-l	Lists all listen sockets.
-v	Lists all virtual servers.
-r	Lists the specified resources

Example

```
HttpServerAdmin list -c -d /export/sun/server6 -sinst
https-sun.com

HttpServerAdmin list -l -d /export/sun/server6 -sinst
https-sun.com
```

The list of information appears in your command window.

Appendix B

Hypertext Transfer Protocol

This appendix provides a short introduction to a few Hypertext Transfer Protocol (HTTP) basics. For more information on HTTP, see the Internet Engineering Task Force (IETF) home page at:

```
http://www.ietf.org/home.html
```

This appendix contains the following sections:

About Hypertext Transfer Protocol (HTTP)

The **Hypertext Transfer Protocol (HTTP)** is a protocol (a set of rules that describe how information is exchanged on a network) that allows a web browser and a web server to “talk” to each other using the ISO Latin1 alphabet, which is ASCII with extensions for European languages.

HTTP is based on a request / response model. The client connects to the server and sends a request to the server. The request contains the following: request method, URI, and protocol version. The client then sends some header information. The server’s response includes the return of the protocol version, status code, followed by a header that contains server information, and then the requested data. The connection is then closed.

The iPlanet Web Server 4.x supports HTTP 1.1. Previous versions of the server supported HTTP 1.0. The server is conditionally compliant with the HTTP 1.1 proposed standard, as approved by the Internet Engineering Steering Group (IESG) and the Internet Engineering Task Force (IETF) HTTP working group. For more information on the criteria for being conditionally compliant, see the Hypertext Transfer Protocol—HTTP / 1.1 specification (RFC 2068) at:

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

Requests

A request from a client to a server includes the following information:

- Request method
- Request header
- Request data

Request Method

A client can request information using a number of methods. The commonly used methods include the following:

- `GET`—Requests the specified document
- `HEAD`—Requests only the header information for the document
- `POST`—Requests that the server accept some data from the client, such as form input for a CGI program
- `PUT`—Replaces the contents of a server's document with data from the client

Request Header

The client can send header fields to the server. Most are optional. Some commonly used request headers are shown in Table B-1.

Table B-1 Common request headers

Request header	Description
`Accept`	The file types the client can accept.

Table B-1 Common request headers *(Continued)*

Request header	Description
`Authorization`	Used if the client wants to authenticate itself with a server; information such as the username and password are included.
`User-agent`	The name and version of the client software.
`Referer`	The URL of the document where the user clicked on the link.
`Host`	The Internet host and port number of the resource being requested.

Request Data

If the client has made a `POST` or `PUT` request, it can send data after the request header and a blank line. If the client sends a `GET` or `HEAD` request, there is no data to send; the client waits for the server's response.

Responses

The server's response includes the following:

- Status code
- Response header
- Response data

Status Code

When a client makes a request, one item the server sends back is a status code, which is a three-digit numeric code. There are four categories of status codes:

- Status codes in the 100–199 range indicate a provisional response.
- Status codes in the 200–299 range indicate a successful transaction.
- Status codes in the 300–399 range are returned when the URL can't be retrieved because the requested document has moved.
- Status codes in the 400–499 range indicate the client has an error.

- Status codes of 500 and higher indicate that the server can't perform the request, or an error has occurred.

Table B-2 contains some common status codes.

Table B-2 Common HTTP status codes

Status code	Meaning
200	OK; successful transmission. This is not an error.
302	Found. Redirection to a new URL. The original URL has moved. This is not an error; most browsers will get the new page.
304	Use a local copy. If a browser already has a page in its cache, and the page is requested again, some browsers (such as Netscape Navigator) relay to the web server the "last-modified" timestamp on the browser's cached copy. If the copy on the server is not newer than the browser's copy, the server returns a 304 code instead of returning the page, reducing unnecessary network traffic. This is not an error.
401	Unauthorized. The user requested a document but didn't provide a valid username or password.
403	Forbidden. Access to this URL is forbidden.
404	Not found. The document requested isn't on the server. This code can also be sent if the server has been told to protect the document by telling unauthorized people that it doesn't exist.
500	Server error. A server-related error occurred. The server administrator should check the server's error log to see what happened.

Response Header

The response header contains information about the server and information about the document that will follow. Common response headers are shown in Table B-3.

Table B-3 Common response headers

Response header	Description
Server	The name and version of the web server.

Table B-3 Common response headers

Response header	Description
`Date`	The current date (in Greenwich Mean Time).
`Last-modified`	The date when the document was last modified.
`Expires`	The date when the document expires.
`Content-length`	The length of the data that follows (in bytes).
`Content-type`	The MIME type of the following data.
`WWW-authenticate`	Used during authentication and includes information that tells the client software what is necessary for authentication (such as username and password).

Response Data

The server sends a blank line after the last header field. The server then sends the document data.

Appendix C

ACL File Syntax

This appendix describes the access-control list (ACL) files and their syntax. ACL files are text files that contain lists that define who can access resources stored on your web server. By default, the web server uses one ACL file that contains all of the lists for access to your server. However, you can create multiple ACL files and reference them in the `obj.conf` file.

You need to know the syntax and function of ACL files if you plan on customizing access control using the access-control API. For example, you might use the access control API to interface with another database, such as an Oracle or Informix database. For more information on the API, see the Sun ONE documentation site at:

```
http://docs.sun.com
```

This appendix contains the following sections:

ACL File Syntax

All ACL files must follow a specific format and syntax. An ACL file is a text file containing one or more ACLs. All ACL files must begin with the version number they use. There can be only one version line and it can appear after any comment lines. Sun ONE Web Server 6.1 uses version 3.0. For example:

```
version 3.0;
```

You can include comments in the file by beginning the comment line with the # sign.

Each ACL in the file begins with a statement that defines its type. ACLs can follow one of three types:

- **Path ACLs** specify an absolute path to the resource they affect
- **URI (Uniform Resource Indicator) ACLs** specify a directory or file relative to the server's document root.
- **Named ACLs** specify a name that is referenced in resources in the `obj.conf` file. The server comes with a "default" named resource that allows read access to anyone and write access to users in the LDAP directory. Even though you can create a named ACL from the Sun ONE Web Server windows, you must manually reference the named ACLs with resources in the `obj.conf` file.

Path and URI ACLs can include wildcards at then end of the entry. For example: `/a/b/*`. Wildcards placed anywhere except at the end of the entry will not work.

The type line begins with the letters `acl` and then includes the type information in double-quotation marks followed by a semicolon. Each type information for all ACLs must be a unique name--even among different ACL files. The following lines are examples of several different types of ACLs:

```
acl "path=C:/sun/Servers/docs/mydocs/";
acl "default";
acl "uri=/mydocs/";
```

After you define the type of ACL, you can have one or more statements that define the method used with the ACL (authentication statements) and the people and computers who are allowed or denied access (authorization statements). The following sections describe the syntax for these statements.

This section includes the following topics:

Authentication Methods

ACLs can optionally specify the authentication method the server must use when processing the ACL. There are three general methods:

- Basic (default)
- Digest

- SSL

Basic and digest require users to enter a username and password before accessing a resource.

SSL requires the user to have a client certificate. The web server must have encryption turned on, and the user's certificate issuer must be in the list of trusted CAs to be authenticated.

By default, the server uses the Basic method for any ACL that doesn't specify a method. Your server's authentication database must be able to handle digest authentication sent by a user.

Each authenticate line must specify what attribute (users, groups, or both users and groups) the server authenticate. The following authentication statement, which would appear after the ACL type line, specifies basic authentication with users matched to individual users in the database or directory:

```
authenticate (user) {
     method = “basic”;
};
```

The following example uses SSL as the authentication method for users and groups:

```
authenticate (user, group) {
     method = “ssl”;
};
```

The following example allows any user whose username begins with the letters `sales`:

```
authenticate (user)

allow (all)

   user = sales*
```

If the last line was changed to `group = sales`, then the ACL would fail because 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. Use the following syntax when writing authorization statements:

```
allow|deny [absolute] (right[,right...]) attribute expression;
```

Start each line with either allow or deny. It's usually a good idea to deny access to everyone in the first rule and then specifically allow access for users, groups, or computers in subsequent rules. This is because of the hierarchy of rules. That is, if you allow anyone access to a directory called `/my_stuff`, and then you have a subdirectory `/my_stuff/personal` that allows access to a few users, the access control on the subdirectory won't work because anyone allowed access to the `/my_stuff` directory will also be allowed access to the `/my_stuff/personal` directory. To prevent this, create a rule for the subdirectory that first denies access to anyone and then allows it for the few users who need access.

However, in some cases if you set the default ACL to deny access to everyone, then your other ACL rules don't need a "deny all" rule.

The following line denies access to everyone:

```
deny (all)
    user = "anyone";
```

This section includes the following topics:

- Hierarchy of Authorization Statements
- Attribute Expressions
- Operators For Expressions

Hierarchy of Authorization Statements

ACLs have a hierarchy that depends on the resource. For example, if the server receives a request for the document (URI) `/my_stuff/web/presentation.html`, the server builds a list of ACLs that apply for this URI. The server first adds ACLs listed in 'check-acl' statements of it's `obj.conf` file. Then the server appends matching URI and PATH ACLs.

The server processes this list 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 there are more than one ACLs that match, the server uses the last statement that matches. However, if you use an absolute statement, then the server stops looking for other matches and uses the ACL containing the absolute statement. If you have two absolute statements for the same resource, the server uses the first one in the file and stops looking for other resources that match.

```
version 3.0;
acl "default";
authenticate (user,group) {
     prompt="Web Server";
};
allow (read,execute,list,info)
     user = "anyone";
allow (write,delete)
     user = "all";
acl "uri=/my_stuff/web/presentation.html";
deny (all)
     user = "anyone";
allow (all)
     user = "joe";
```

Attribute Expressions

Attribute expressions define who is allowed or denied access based on their username, group name, host name, or IP address. The following lines are examples of allowing access to different people or computers:

- `user = “anyone”`
- `user = “smith*”`
- `group = “sales”`
- `dns = “*.sun.com”`
- `dns = “*.sun.com,*.mozilla.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.

NOTE Use 24-hour time to specify times. For example, use 0400 to specify 4:00 a.m. or 2230 for 10:30 p.m.

The following example restricts access to a group of users called guests between 8:00 a.m. and 4:59 p.m:

```
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 get access all day on weekends and on weekdays anytime except 8am-4:59pm.

```
allow (read) (group="discount" and dayofweek="Sat,Sun") or
(group="discount" and (dayofweek="mon,tue,wed,thu,fri" and
(timeofday<0800 or timeofday=1700)))
or
(group="premium");
```

Operators For Expressions

You can use various operators in attribute expressions. Parentheses delineate the operator order of precedence. With `user`, `group`, `dns`, and `ip`, you can use the following operators:

- and
- or
- not
- = (equals)
- != (not equal to)

With `timeofday` and `dayofweek`, you can use:

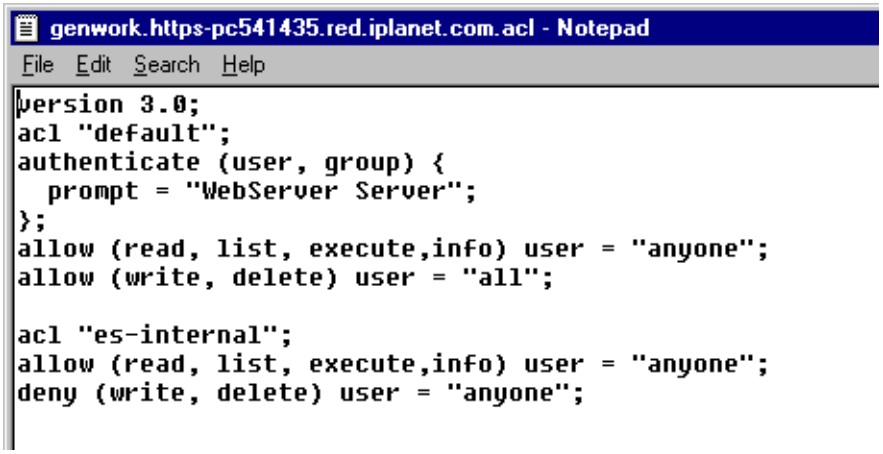
- greater than
- < less than

- = greater than or equal to
- <= less than or equal to

The Default ACL File

After installation, the *server_root*/httpacl/generated.https-*serverid*.acl file provided default settings for the server. The server uses the working file genwork.https-*serverid*.acl until you create settings in the user interface. When editing an ACL file, you could make changes in the genwork file, then save and apply the changes using Sun ONE Web Server.

genwork File.



General Syntax Items

Input strings can contain the following characters:

- Letters a through z
- Numbers 0 through 9
- Period and underscore

If you use any other characters, you need to use double-quotation marks around the characters.

A single statement can be placed on its own line and be terminated with a semicolon. Multiple statements are placed within braces. A list of items must be separated by commas and enclosed in double-quotation marks.

Referencing ACL Files in obj.conf

If you have named ACLs or separate ACL files, you can reference them in the `obj.conf` file. You do this in the `PathCheck` directive using the `check-acl` function. The line has the following syntax:

```
PathCheck fn="check-acl" acl="aclname"
```

The `aclname` is a unique name of an ACL as it appears in any ACL file.

For example, you might add the following lines to your `obj.conf` file if you want to restrict access to a directory using the ACL named `testacl`:

```
<Object ppath="/usr/ns-home/docs/test/*"
PathCheck fn="check-acl" acl="testacl"
</Object
```

In the previous example, the first line is the object that states which server resource you want to restrict access to. The second line is the `PathCheck` directive that uses the `check-acl` function to bind the name ACL (`testacl`) to the object in which the directive appears. The `testacl` ACL can appear in any ACL file referenced in `magnus.conf`.

Appendix D

Support for Internationalization and Localization

The internationalized and localized version of the Sun ONE Web Server 6.1 provides support for multiple languages and multiple encodings.

The main features are described in this appendix:

Entering Multibyte Data

If you want to enter multibyte data on the Server Manager or the Administration Server pages, you need to be aware of the following issues:

File or Directory Names

If a file or directory name is to appear in a URL, it cannot contain 8-bit or multibyte characters.

LDAP Users and Groups

For email addresses, use only those characters permitted in RFC 1700 (`ftp://ds.internic.net/rfc/rfc1700.txt`). User ID and password information must be stored in ASCII.

To make sure you enter characters in the correct format for users and groups, use a `UTF-8` form-capable client (such as Netscape Communicator) to input 8-bit or multibyte data.

Support for Multiple Character Encodings

Sun ONE Web Server 6.1 provides multiple character encoding support for the following features:

- WebDAV
- Search

WebDAV

Sun ONE Web Server 6.1 supports setting and retrieving multibyte properties in the `PROPPPATCH` and `PROPFIND` methods. While request can be in any encoding format, the response from the server is always in UTF-8.

Search

Sun ONE Web Server 6.1 uses a Java-based search engine that supports full-text indexing and searching of documents in all character encodings that the underlying Java VM supports. The default encoding for the documents can be specified at the time of creating a search collection. For HTML documents, the indexer tries to deduce the encoding from the HTML metatags and if it cannot, falls back to use the default encoding.

The search interface is based on JSP tag libraries and can be customized and localized in any language and encoding that you wish. The tag libraries are listed in the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*. For more information, see "Customizing the Search Query Page" on page 409.

Language Preferences

You can set the Default Language for your server which is used for all the end user error messages using the Magnus Editor from the server preferences. The localized version of Sun ONE Web Server 6.1 supports seven languages:

- en (English)
- fr (French)
- de (German)
- ja (Japanese)
- ko (Korean)
- zh (Simplified Chinese)
- zh_TW (Traditional Chinese)

The end-user search interface in a localized version of the Sun ONE Web Server 6.1 is completely localized.

NOTE This setting has no effect on a non-localized version of the web server.

Configuring the Server to Serve Localized Content

End users can configure their browsers to send an Accept-language header that describes their language preference for the content they are accessing. The server can be configured to serve content based on the Accept-language header by turning the `acceptlanguage` setting on for the `VS` class in the Edit Classes menu of the Administration Server. This also ensures that all end user error messages are also based on the Accept-language header.

For example, if `acceptlanguage` is set to `on`, and a client sends the Accept-language header with the value `fr-CH,de`, when requesting the following URL:

```
http://www.someplace.com/somepage.html
```

Your server searches for the file in the following order:

1. The `Accept-language` list `fr-CH, de`.

   `http://www.someplace.com/fr_ch/somepage.html`

   `http://www.someplace.com/somepage_fr_ch.html`

   `http://www.someplace.com/de/somepage.html`

   `http://www.someplace.com/somepage_de.html`

2. Language codes without the country codes (`fr` in the case of `fr-CH`):

   `http://www.someplace.com/fr/somepage.html`

   `http://www.someplace.com/somepage_fr.html`

3. The `DefaultLanguage`, such as `en`, defined in the `magnus.conf` file.

   `http://www.someplace.com/en/somepage.html`

   `http://www.someplace.com/somepage_en.html`

4. If none of these are found, the server tries:

   `http://www.someplace.com/somepage.html`

NOTE Keep in mind when naming your localized files that country codes like CH and TW are converted to lower case and dashes (-) are converted to underscores (_).

CAUTION Enabling the `acceptlanguage` setting has a performance penalty since the server has to check for content in every language specified in the Accept-language as per the algorithm illustrated above.

Glossary

Access Control Entries (ACEs) A hierarchy of rules which the web server uses to evaluate incoming access requests.

Access Control List (ACL) A collection of ACEs. An ACL is a mechanism for defining which users have access to your server. You can define ACL rules that are specific to a particular file or directory, granting or denying access to one or more users and groups.

admpw The username and password file for the Enterprise Administrator Server superuser.

agent Software that runs the network-management software in a network device, such as a router, host, or X terminal. See also intelligent agents.

authentication Allows clients to verify their identity to the server. Basic or Default authentication requires users to enter a username and password to access your web server or web site. It requires a list of users and groups in an LDAP database. See also digest and SSL authentication.

The granting of access to an entire server or particular files and directories on it. Authorization can be restricted by criteria including hostnames and IP addresses.

browser See client.

cache A copy of original data that is stored locally. Cached data doesn't have to be retrieved from a remote server again when requested.

certificate A nontransferable, nonforgeable, digital file issued from a third party that both communicating parties already trust.

certification authority (CA) An internal or third-party organization that issues digital files used for encrypted transactions.

Certificate revocation list (CRL) CA list, provided by the CA, of all revoked certificates.

Compromised key list (CKL) A list of key information about users who have compromised keys. The CA also provides this list.

CGI Common Gateway Interface. An interface by which external programs communicate with the HTTP server. Programs that are written to use CGI are called CGI programs or CGI scripts. CGI programs handle forms or parse output the server does not normally handle or parse.

chroot An additional root directory you can create to limit the server to specific directories. You'd use this feature to safeguard an unprotected server.

cipher A cipher is a cryptographic algorithm (a mathematical function), used for encryption or decryption.

ciphertext Information disguised by encryption, which only the intended recipient can decrypt.

client Software, such as Netscape Navigator, used to request and view World Wide Web material. Also known as a browser program.

client auth Client authentication.

cluster A group of remote 'slave' administration servers added to and controlled by a 'master' and administration server. All servers in a cluster must be of the same platform and have the same userid and password.

collection A database that contains information about documents, such as word list and file properties. Collections are used by the search function to retrieve documents matching specified search criteria.

Common LogFile Format The format used by the server for entering information into the access logs. The format is the same among all major servers, including the Sun ONE Web Server.

DHCP Dynamic Host Configuration Protocol. An Internet Proposed Standard Protocol that allows a system to dynamically assign an IP address to individual computers on a network.

daemon (UNIX) A background process responsible for a particular system task.

digest authentication. Allows the user to authenticate without sending the username and password as cleartext. The browser uses the MD5 algorithm to create a digest value. The server uses the Digest Authentication plug-in to compare the digest value provided by the client.

DNS Domain Name System. The system that machines on a network use to associate standard IP addresses (such as 198.93.93.10) with hostnames (such as `www.sun.com`). Machines normally get this translated information from a DNS server, or they look it up in tables maintained on their systems.

DNS alias A hostname that the DNS server knows points to a different host—specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, an alias such as `www.`*yourdomain.domain* might point to a real machine called `realthing.`*yourdomain.domain* where the server currently exists.

document root A directory on the server machine that contains the files, images, and data you want to present to users accessing the server.

drop word See stop word.

encryption The process of transforming information so it can't be decrypted or read by anyone but the intended recipient.

Administration Server A web-based server that contains the forms you use to configure all of your Sun ONE Web Servers.

expires header The expiration time of the returned document, specified by the remote server.

extranet An extension of a company's intranet onto the Internet, to allow customers, suppliers, and remote workers access to the data.

file extension The last part of a filename that typically defines the type of file. For example, in the filename `index.html` the file extension is `html`.

file type The format of a given file. For example, a graphics file doesn't have the same file type as a text file. File types are usually identified by the file extension (`.gif` or `.html`).

firewall A network configuration, usually both hardware and software, that protects networked computers within an organization from outside access. Firewalls are commonly used to protect information such as a network's email and data files within a physical building or organization site.

flexible log format A format used by the server for entering information into the access logs.

FORTEZZA An encryption system used by U.S. government agencies to manage sensitive but unclassified information.

FTP File Transfer Protocol. An Internet protocol that allows files to be transferred from one computer to another over a network.

GIF Graphics Interchange Format. A cross-platform image format originally created by CompuServe. GIF files are usually much smaller in size than other graphic file types (BMP, TIFF). GIF is one of the most common interchange formats. GIF images are readily viewable on UNIX, Microsoft Windows, and Apple Macintosh systems.

hard restart The termination of a process or service and its subsequent restart. See also soft restart.

home page A document that exists on the server and acts as a catalog or entry point for the server's contents. The location of this document is defined within the server's configuration files.

hostname A name for a machine in the form *machine.domain.dom*, which is translated into an IP address. For example, `www.sun.com` is the machine `www` in the subdomain `sun` and `com` domain.

HTML Hypertext Markup Language. A formatting language used for documents on the World Wide Web. HTML files are plain text files with formatting codes that tell browsers such as the Netscape Navigator how to display text, position graphics and form items, and display links to other pages.

HTTP HyperText Transfer Protocol. The method for exchanging information between HTTP servers and clients.

HTTP-NG The next generation of HyperText Transfer Protocol.

HTTPD An abbreviation for the HTTP daemon or service, a program that serves information using the HTTP protocol. The Sun ONE Web Server is often called an HTTPD.

HTTPS A secure version of HTTP, implemented using the Secure Sockets Layer, SSL.

imagemap A process that makes areas of an image active, letting users navigate and obtain information by clicking the different regions of the image with a mouse. Imagemap can also refer to a CGI program called "imagemap," which is used to handle imagemap functionality in other HTTPD implementations.

inittab (UNIX) A UNIX file listing programs that need to be restarted if they stop for any reason It ensures that a program runs continuously. Because of its location, it is also called `/etc/inittab`. This file isn't available on all UNIX systems.

intelligent agent An object within a server that performs various requests (such as HTTP, NNTP, SMTP, and FTP requests) on behalf of the user. In a sense, the intelligent agent acts as a client to the server, making requests that the server fulfills.

IP address Internet Protocol address. A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 198.93.93.10).

ISDN Integrated Services Digital Network.

ISINDEX An HTML tag that turns on searching in the client. Documents can use a network navigator's capabilities to accept a search string and send it to the server to access a searchable index without using forms. In order to use <ISINDEX>, you must create a query handler.

ISMAP ISMAP is an extension to the IMG SRC tag used in an HTML document to tell the server that the named image is an imagemap.

ISP Internet Service Provider. An organization that provides Internet connectivity.

Java An object-oriented programming language created by Sun Microsystems used to create real-time, interactive programs called applets.

JavaScript A compact, object-based scripting language for developing client and server Internet applications.

JavaServer Pages Extensions that enable all JavaServer page metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. JavaServer pages, are reusable Java applications that run on a web server rather than in a web browser.

Java Servlets Extensions that enable all Java servlet metafunctions, including instantiation, initialization, destruction, access from other components, and configuration management. Java servlets are reusable Java applications that run on a web server rather than in a web browser.

last-modified header The last modification time of the document file, returned in the HTTP response from the server.

LDAP database A database where lists of users and groups is stored for use in authentication.

listen socket The combination of port number and IP address. Connections between the server and clients happen on a listen socket.

magnus.conf The main Web Server configuration file. This file contains global server configuration information (such as, port, security, and so on). This file sets the values for variables that configure the server during initialization. Enterprise Sever reads this file and executes the variable settings on startup. The server does not read this file again until it is restarted, so you must restart the server every time you make changes to this file.

MD5 A message digest algorithm by RSA Data Security. MD5 can be used to produce a short digest of data that is unique with high probability. It is mathematically extremely hard to produce a piece of data that produces the same message digest email.

MD5 signature A message digest produced by the MD5 algorithm.

MIB Management Information Base.

MIME Multi-Purpose Internet Mail Extensions. An emerging standard for multimedia email and messaging.

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

modutil Software utility required for installing PKCS#11 module for external encryption or hardware accelerator devices.

MTA Message Transfer Agent. You must define your server's MTA Host to use agent services on your server.

NIS (UNIX) Network Information Service. A system of programs and data files that UNIX machines use to collect, collate, and share specific information about machines, users, file systems, and network parameters throughout a network of computers.

network management station (NMS) A machine users can use to remotely manage a network. A managed device is anything that runs SNMP such as hosts, routers, and Sun ONE servers. An NMS is usually a powerful workstation with one or more network management applications installed.

NNTP Network News Transfer Protocol for newsgroups. You must define your news server host to use agent services on your server.

NSAPI See Server Plug-in API.

obj.conf The server's object configuration file. This file contains additional initialization information, settings for server customization, and instructions that the server uses to process requests from clients (such as browsers). Sun ONE Web Server reads this file every time it processes a client request.

password file (UNIX) A file on UNIX machines that stores UNIX user login names, passwords, and user ID numbers. It is also known as `/etc/passwd`, because of where it is kept.

pk12util Software utility required to export the certificate and key databases from your internal machine, and import them into an external PKCS#11 module.

primary document directory See document root.

protocol A set of rules that describes how devices on a network exchange information.

private key The decryption key used in public-key encryption.

public key The encryption key used in public-key encryption.

public information directories (UNIX) Directories not inside the document root that are in a UNIX user's home directory, or directories that are under the user's control.

Quality of Service the performance limits you set for a server instance, virtual server class, or virtual server.

RAM Random access memory. The physical semiconductor-based memory in a computer.

rc.2.d (UNIX) A file on UNIX machines that describes programs that are run when the machine starts. This file is also called `/etc/rc.2.d` because of its location.

redirection A system by which clients accessing a particular URL are sent to a different location, either on the same server or on a different server. This system is useful if a resource has moved and you want the clients to use the new location transparently. It's also used to maintain the integrity of relative links when directories are accessed without a trailing slash.

resource Any document (URL), directory, or program that the server can access and send to a client that requests it.

RFC Request For Comments. Usually, procedures or standards documents submitted to the Internet community. People can send comments on the technologies before they become accepted standards.

root (UNIX) The most privileged user on UNIX machines. The root user has complete access privileges to all files on the machine.

server daemon A process that, once running, listens for and accepts requests from clients.

Server Plug-in API An extension that allows you to extend and/or customize the core functionality of Sun ONE servers and provide a scalable, efficient mechanism for building interfaces between the HTTP server and back-end applications. Also known as NSAPI.

server root A directory on the server machine dedicated to holding the server program, configuration, maintenance, and information files.

SOCKS Firewall software that establishes a connection from inside a firewall to the outside when direct connection would otherwise be prevented by the firewall software or hardware (for example, the router configuration).

soft restart A way to restart the server that causes the server to internally restart, that is, reread its configuration files. A soft restart sends the process the HUP signal (signal number one). The process itself does not die, as it does in a hard restart.

SSL Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

SSL authentication Confirms users' identities with security certificates by using the information in the client certificate as proof of identity, or verifying a client certificate published in an LDAP directory.

stop word A word identified to the search function as a word not to search on. This typically includes such words as the, a, an, and. Also referred to as drop words.

strftime A function that converts a date and a time to a string. It's used by the server when appending trailers. `strftime` has a special format language for the date and time that the server can use in a trailer to illustrate a file's last-modified date.

Sun ONE Web Server Administration Console Formerly, known as Netscape Console, this is a Java application that provides server administrators with a graphical interface for managing all Sun ONE servers from one central location anywhere within your enterprise network. From any installed instance of the Sun ONE Web Server Administration Console, you can see and access all the Sun ONE servers on your enterprise's network to which you have been granted access rights.

superuser (UNIX) The most privileged user available on UNIX machines (also called root). The superuser has complete access privileges to all files on the machine.

Sym-links (UNIX) Abbreviation for symbolic links, which is a type of redirection used by the UNIX operating system. Sym-links let you create a pointer from one part of your file system to an existing file or directory on another part of the file system.

TCP/IP Transmission Control Protocol/Internet Protocol. The main network protocol for the Internet and for enterprise (company) networks.

telnet A protocol where two machines on the network are connected to each other and support terminal emulation for remote login.

timeout A specified time after which the server should give up trying to finish a service routine that appears hung.

TLS Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

top (UNIX) A program on some UNIX systems that shows the current state of system resource usage.

top-level domain authority The highest category of hostname classification, usually signifying either the type of organization the domain is (for example, `.com` is a company, `.edu` is an educational institution) or the country of its origin (for example, `.us` is the United States, `.jp` is Japan, `.au` is Australia, `.fi` is Finland).

uid (UNIX) A unique number associated with each user on a UNIX system.

URI Uniform Resource Identifier. A file identifier that provides an additional layer of security by using an abbreviated URL. The first part of the URL is substituted with a URL mapping that hides the file's full physical pathname from the user. See also URL mapping.

URL Uniform Resource Locator. The addressing system used by the server and the client to request documents. A URL is often called a location. The format of a URL is *protocol://machine:port/document*.

A sample URL is `http://www.sun.com/index.html`.

URL database repair A process that repairs and updates a URL database that has been damaged by a software failure, a system crash, a disk breakdown, or a full file system.

URL mapping The process of mapping a document directory's physical pathname to a user-defined alias so that files within the directory need only refer to the directory's alias instead of the file's full physical pathname. Thus, instead of identifying a file as `usr/sun/servers/docs/index.html`, you could identify the file as `/myDocs/index.html`. This provides additional security for a server by eliminating the need for users to know the physical location of server files.

virtual server class A collection of virtual servers that shares the same basic configuration information in a obj.conf file.

virtual server Virtual servers are a way of setting up multiple domain names, IP addresses, and server monitoring capabilities with a single installed server.

web application A collection of servlets, JavaServer Pages, HTML documents, and other web resources which might include image files, compressed archives, and other data. A web application may be packaged into an archive (a WAR file) or exist in an open directory structure.

Web Application Archive (WAR) An archive file that contains a complete web application in compressed form. Sun ONE Web Server cannot access an application in a WAR file. You must uncompress a web application (deploy it using the `wdeploy` utility) before Sun ONE Web Server can serve it.

Windows CGI (Windows) CGI programs written in a Windows-based programming language such as Visual Basic.

Index

SYMBOLS

NUMERICS

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X