Sun Java System Web Server 7.0 Update 4 Administrator's Guide

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

Part No: 820–6600
December 2008

090324@21990

Contents

Figures

Tables

Examples

Preface

This guide describes how to configure and administer the *Sun Java™ System Web Server 7.0 Update 4* (also referred to as Web Server).

Who Should Use This Book

This book is intended for Sun Java System Web Server administrators to manage the server in production environments. The guide assumes familiarity with the following areas:

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

Web Server Documentation Set

The Web Server documentation set describes how to install and administer the Web Server. You can access Web Server Update 4 documentation at `http://docs.sun.com/coll/1653.4`.

The Sun Java System Web Server documents are now in wiki format at `http://wikis.sun.com/display/WebServerdocs/Home`. This wiki is intended to promote collaboration and contribution on documentation content for Web Server. You are welcome to contribute, by posting your comments or by directly editing the wiki page, as long as the content is relevant to an appropriate standard.

For an introduction to Web Server Update 4, refer to the books in the order in which they are listed in the following table.

TABLE P–1 Books in the Web Server Documentation Set

Documentation Title	Contents
Sun Java System Web Server 7.0 Update 4 Documentation Center	Web Server documentation topics organized by tasks and subject

TABLE P–1 Books in the Web Server Documentation Set *(Continued)*

Documentation Title	Contents
Sun Java System Web Server 7.0 Update 4 Release Notes	■ Late-breaking information about the software and documentation ■ Supported platforms and patch requirements for installing Web Server
Sun Java System Web Server 7.0 Update 4 Installation and Migration Guide	Performing installation and migration tasks: ■ Installing Web Server and its various components, ■ Migrating data from Sun ONE Web Server 6.0 or 6.1 to Sun Java System Web Server 7.0
Sun Java System Web Server 7.0 Update 4 Administrator's Guide	Performing the following administration tasks: ■ Using the Administration GUI and command-line interface ■ 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 Platform Enterprise Edition (Java EE) 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
Sun Java System Web Server 7.0 Update 4 Developer's Guide	Using programming technologies and APIs to do the following: ■ Extend and modify Sun Java System Web Server ■ Dynamically generate content in response to client requests and modify the content of the server
Sun Java System Web Server 7.0 Update 4 NSAPI Developer's Guide	Creating custom Netscape Server Application Programmer's Interface (NSAPI) plug-ins
Sun Java System Web Server 7.0 Update 4 Developer's Guide to Java Web Applications	Implementing Java Servlets and JavaServer Pages™ (JSP™) technology in Sun Java System Web Server

TABLE P–1 Books in the Web Server Documentation Set *(Continued)*

Documentation Title	Contents
Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference	Editing configuration files
Sun Java System Web Server 7.0 Update 4 Performance Tuning, Sizing, and Scaling Guide	Tuning Sun Java System Web Server to optimize performance
Sun Java System Web Server 7.0 Update 4 Troubleshooting Guide	Troubleshooting Web Server

Related Books

The URL for all documentation about Sun Java Enterprise System (Java ES) and its components is `http://docs.sun.com/coll/1286.2`.

Default Paths and File Names

The following table describes the default paths and file names that are used in this book.

TABLE P–2 Default Paths and File Names

Placeholder	Description	Default Value

install-dir	Represents the base installation directory for Web Server	Sun Java Enterprise System (Java ES) installations on the Solaris™ platform:
		`/opt/SUNWwbsvr7`
		Java ES installations on the Linux and HP-UX platform:
		`/opt/sun/webserver/`
		Java ES installations on the Windows platform:
		system-drive`:\Program Files\Sun\JavaES5\WebServer7`
		Other Solaris, Linux, and HP-UX installations, non-root user:
		home-directory`/sun/webserver7`
		Other Solaris, Linux, and HP-UX installations, root user:
		`/sun/webserver7`
		Windows, all installations:
		system-drive`:\Program Files\Sun\WebServer7`

TABLE P–2 Default Paths and File Names *(Continued)*

Placeholder	Description	Default Value

instance-dir	Directory that contains the instance-specific subdirectories.	For Java ES installations, the default location for instances on Solaris:
		`/var/opt/SUNWwbsvr7`
		For Java ES installations, the default location for instances on Linux and HP-UX:
		`/var/opt/sun/webserver7`
		For Java ES installations, the default location for instance on Windows:
		system-drive`:\Program Files\Sun\JavaES5\WebServer7`
		For stand-alone installations, the default location for instance on Solaris, Linux, and HP-UX:*install-dir*
		For stand-alone installations, the default location for instance on Windows:
		system-drive`:\Program Files\sun\WebServer7`

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–3 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	A placeholder to be replaced with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file.

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–4 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Accessing Sun Resources Online

The `http://docs.sun.com` (docs.sun.com[SM]) web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.com web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for "Web Server," type the following:

```
Web Server site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use "`sun.com`" in place of "`docs.sun.com`" in the search field.

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

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

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments:

- Go to `http://docs.sun.com` and click Feedback.
- Go to `http://wikis.sun.com/display/WebServerdocs/Home` and post your comments or directly edit the wiki page.

◆ ◆ ◆ CHAPTER 1

Getting Started

This chapter describes the basics of Sun Java System Web Server 7.0 by providing a brief description of terms used in this guide.

Introduction

Web Server is a multi-process, multi-threaded, secure web server built on industry standards. It provides high performance, reliability, scalability, and manageability for medium to large enterprises.

Web Server provides comprehensive command-line interface support, consolidated configuration, enhanced security with Elliptic Curve Cryptography support, and clustering support. It also comes with a robust built-in migration tool that helps migrate applications and configurations from Web Server 6.0 and 6.1 to Web Server .

What is New?

See Chapter 1, "Sun Java System Web Server Release Notes," in *Sun Java System Web Server 7.0 Update 4 Release Notes* for more information on the new features and enhancements in Sun Java System Web Server 7.0.

Starting the Administration Server

In order to use the administration interface, you need to start the Administration Server.

Starting the Administration Server in Unix/Linux

To start the Administration Server perform the following tasks:

▼ Starting the Administration Server in Unix/Linux

1 **Go to the** `install_root/admin-server/bin` **directory (for example,** `/usr/sjsws7.0/admin-server/bin`**)**

2 **Type** `./startserv`**.**

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

Starting the Administration Server in Windows

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

- Release Notes
- Start Administration Server
- Uninstall 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.

Different Ways of Administering Your Server

You can manage Web Server by using the following user interfaces:

- Administration Console (GUI).
- Command Line Interface (`wadm` shell).

You can either use the `wadm` shell interface which is discussed later in this chapter, or the web-based Administration Console to manage instances. Note that the Administration Node can have only one instance of a particular Configuration running.

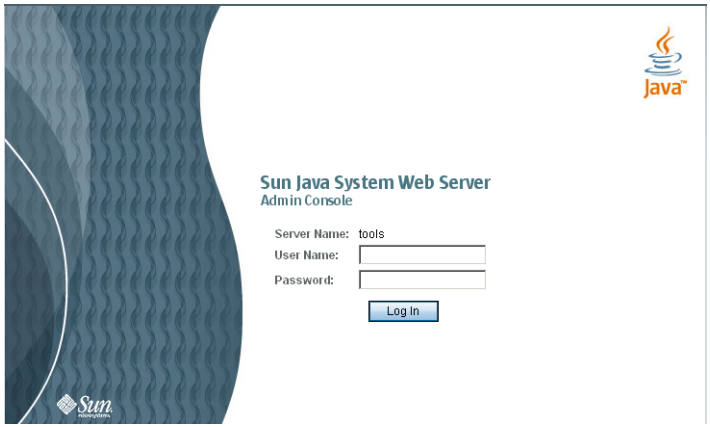
Using Administration Console

After installing Web Server , use your browser to access the Administration Console.

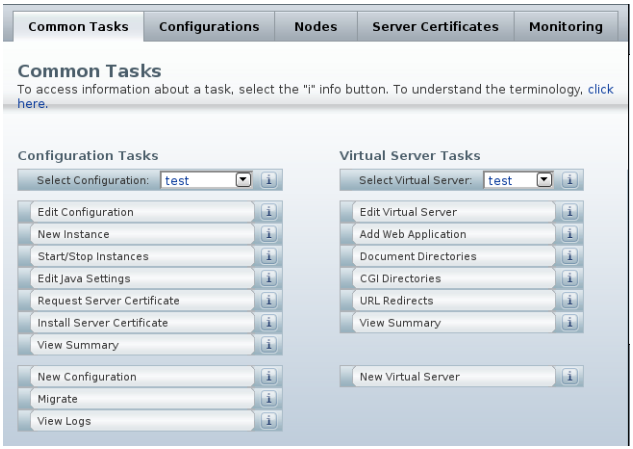
The URL you use to navigate to the Administration Server page depends on the computer host name and the port number you choose for the Administration Server when you install Web Server . For example, if you installed the Administration Server with SSL port 1234, the URL will look like this:

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

You need to log in to the Administration Console to perform server administration. You set up the administrator user name and password when you install Web Server on your computer. The following figure shows the authentication screen:

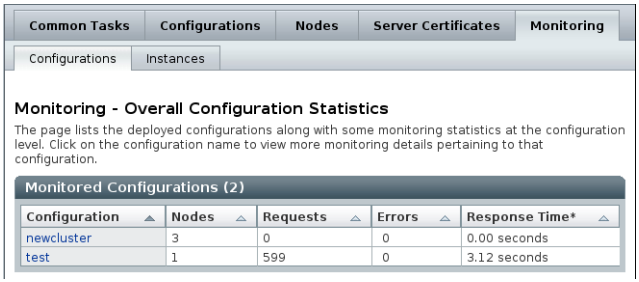


The first page you see when you access the Administration Server is the common tasks page. Use the buttons on this page to manage, add, remove, and migrate servers. The common tasks page is shown in the following figure.



Note – Clicking any of these tabs may result in child tabs appearing on the page. The actions provided by the child tabs are specific to the parent tab functionality.

The following figure shows the child tabs for a selected tab:



Clicking on the tab opens pages in the same window. There are certain tasks that involve gathering data from the user in series of steps. The Administration Console has a wizard interface for such tasks. Wizards always openin a new window.

Help on Administration Console GUI Screens

All form elements and GUI components have a detailed inline help that provides information on the validation and optional parameters. When you use the wizard interface, you can click on the help tab at any time in order to obtain help for the current task.

Using CLI

This section describes the Command Line Interface for Web Server and defines all the commands that are supported for configuring and administering the server.

Web Server has introduced a new CLI called `wadm`.

The earlier version of the Server supported a few discrete command lines, which together addressed only a subset of whole administration functionality provided in the GUI. The command line interfaces supported in Web Server 6.1 were `HttpServerAdmin`, `wdeploy` and `flexanlg`. The new CLI (`wadm`) features include:

- Embedded JACL shell for scripting.
- Extensible CLI — more commands can be added to the CLI by third party plug-ins.

Note – Web Server does not support `HttpServerAdmin`.

Note – `wdeploy` is supported in Web Server only for backward compatibility with 6.x versions and will work only on the Administration Server node.

Modes of CLI

`wadm` supports invocation in the following three modes. They are:

- **Standalone mode** — In this mode, you invoke `wadm` from a command shell, specifying the desired command, options and operands. When the command finishes execution, CLI exits back to the shell. This mode can support both interactive and non-interactive execution of commands. Interactive execution, which is the default, will prompt you for the password if the password is not already specified in the password file and if it has not already passed through the `--password-file` option. Non-interactive execution will result in an error if the `--password-file` option is not specified.

  For example,

  - Non-interactive Standalone mode

```
bash-3.00# cat /passwd
wadm_password=mypassword


bash-3.00# /opt/sun7ur2websvr/bin/wadm list-configs --user=admin
 --port=8800 --no-ssl --password-file=/passwd

instance1
```

- Interactive Standalone mode

```
bash-3.00# /opt/sun7ur2websvr/bin/wadm list-configs --user=admin
 --port=8800 --no-ssl

Please enter admin-user-password>

instance1
```

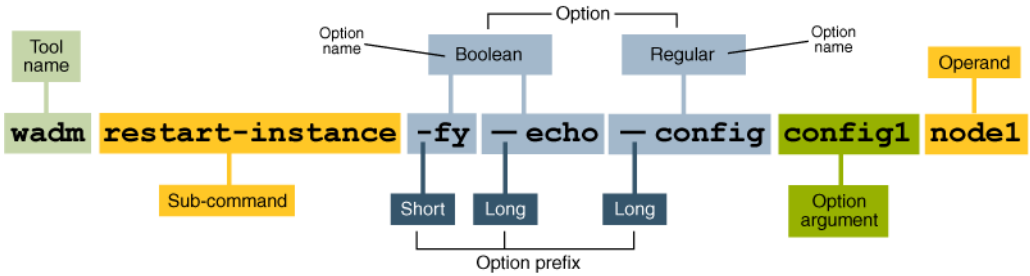
- **Shell Mode** — In this mode, you invoke `wadm` from a command shell with no command. `wadm` prompts the user for a command. After the command is executed, it will return back to the shell. This shell can be exited by typing exit or quit command. Interactive and non-interactive executions are applicable to this mode. For example,

```
bash-3.00# /opt/sun7ur2websvr/bin/wadm --user=admin
 --port=8800 --host=serverhost --no-ssl

Please enter admin-user-password>

Connected to serverhost:8800
Sun Java System Web Server 7.0U2 B12/09/2007 07:28

wadm> list-configs
instance1

wadm> list-jvm-options --config=instance1
-Djava.security.auth.login.config=login.conf
-Xms128m -Xmx256m

wadm> list-instances --config=instance1
sunhost1.sun.com

wadm> quit

bash#
```

- **File Mode** — In this mode, you can add a list of commands in a file and pass the file as an argument to `wadm`. For example,

```
wadm -user=admin -host=serverhost --password-file=admin.pwd
--port=8989 -commands-file=/space/scripts/admscr
```

The following figure depicts the syntax for invoking `wadm` commands.



Note – With the `wadm` CLI, you can perform all of the same tasks as the Administration Console.

Where Can I Find `wadm` CLI?

Question: Where can I find the CLI for Web Server administration?

Answer: The administration CLI is located at `install-root/bin/wadm`. To use the CLI, you need to know:

- Administration server hostname (default is `localhost`).
- SSL port for the administration server (default is 8989).
- Administration server user name (default is `admin`).
- Administration server password.

Note – The Administration server needs to be running in order to use the CLI. You can start the server by running `install-root/admin-server/bin/startserv`.

Authentication in CLI

`wadm` will use the username and password of the administrator to authenticate the Administration Server. A valid username and password file must be passed as arguments to each command running in single mode. The shell mode accepts the username and password file when the `wadm` executable is invoked. Commands invoked in the shell mode do not require the connection options (for example, `user`, `password-file`, `host`, `port` and `ssl`). If connection options are specified, they will be ignored.

Some commands supported by the CLI require password inputs. For example, `bindpw`, `user-password` and `token-pin`. The user can specify these passwords in the same file that contains the administration user password. If the `password-file` is not specified with the command, then user will be prompted for the password.

The `wadm` communicates with the Administration Server through SSL if SSL is enabled on Administration Server. The Certificate passed by the Administration Server will be verified against the *truststore (~/.wadmtruststore)*. If the certificate exists and is valid, the command proceeds normally. Otherwise, `wadm` displays the certificate and gives the user the choice of accepting it. If the user accepts it, the certificate will be added to the *truststore* and the command will proceed normally.

Note – *truststore* need not be password protected since it does not contain any sensitive data.

Resetting the Administration Password

Open a command prompt terminal and navigate to `install-root/bin/wadm` directory and type the following command to reset the password:

```
./wadm reset-admin-password.
```

You will be prompted to type the new admin password. Type the password again to confirm. Restart the server in order for the password to take effect. Then use the new password to login to the server.

Note – While changing the password using `set-admin-prop` command you must restart the Administration Server from `install-root/admin-server/bin/restart` directory. Do not use `restart-admin` command to restart the server.

Registering with Sun Connection

You can use the Admin Console to register the Web Server with Sun Connection. Click the **Register with Sun Connection** tab from the home page, and a wizard opens up. Follow the wizard to complete the registration. By registering the Web Server with Sun Connection you receive the following benefits.

- Patch information and bug updates
- News and events
- Support and training offerings

CLI Scripts

`install-root/samples/admin/scripts` directory contains scripts that you can run using the `wadm` command line utility. `wadm` is built on a TCLengine and hence supports TCL scripting. These scripts can be used to perform common administrative tasks. They also demonstrate how new utilities can be built on top of existing commands.

The following table describes the scripts:

TABLE 1–1 Sample CLI Scripts

Script	Description	Usage
`enable-ssl.tcl`	Enables SSL on a given virtual server and port.	`wadm -f enable-ssl.tcl <config> <vs> <server> <port>`
`filter-mime.tcl`	Fetches the matching MIME types from the given configuration and virtual server.	`wadm -f filter-mime.tcl "<regex>" <config> <vs>`
`remove-mime.tcl`	Removes the matching MIME types from the given configuration and virtual server.	`wadm -f remove-mime.tcl "<regex>" <config> <vs>`
`add-mime-ext.tcl`	Adds the specified extension to the matching MIME types in the given configuration and virtual server.	`wadm -f add-mime-ext.tcl "<regex>" "ext" <config> <vs>`
`summary.tcl`	Provides a summary of the installation. It contains list of listeners, ports, and SSL status.	`wadm -f summary.tcl`
`list-webapps.tcl`	Provides a summary of all the deployed web applications.	`wadm -f list-webapps.tcl`
`collate-logs.tcl`	Provides a collated logs across multiple nodes.	`wadm -f collate-logs.tcl <config> <node1> <node2> ..`
`renew-selfsigned-cert.tcl`	Enables renewal of self-signed certificates with a given nickname	`wadm -f renew-selfsigned-cert.tcl <config> <cert-nickname> [<validity>]`

Understanding Web Server 7.0

Web Server includes a new administration framework that provides enhanced distributed management across servers in a server farm. Robust administration capabilities enable Web Servers to be managed and deployed remotely using both graphical and command-line interfaces. Servers can be managed on a central location in a server farm and distributed to one or more nodes to create server instances. Monitoring and lifecycle management of these server instances are also provided.

Web Server is configured to enable you to turn various features on or off, 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) that identifies these options are stored in configuration files. 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 Web Server *Administrator's Configuration File Reference Guide*.

In Web Server all configurable elements of a server instance like web applications, configuration files, and search collection indexes are logically grouped and termed as a **Configuration**. A Configuration can be created, modified or deleted using CLI or the web based administration interface. You can manage more then one Configuration at a time. The term Configuration also refers to the set of metadata that configures the runtime services of the server. For example, a runtime service serves web pages from a configured document root. The configuration metadata is used by the server runtime to load built-in services, third party plug-ins and setup other server extensions such as database drivers for serving web pages and dynamic web applications.

Note – All the Configuration related files are stored in a repository in your file system called as **Configuration Store**. You must refrain from manually editing any of the files in this repository unless explicitly specified in this guide.

In Web Server, any change to the Configuration using the CLI or through the web-based administration interface is first made to the Configuration Store and then the Configuration is deployed. Consequently the changes are copied to the instance directory. When a web application is deployed it gets deployed under:

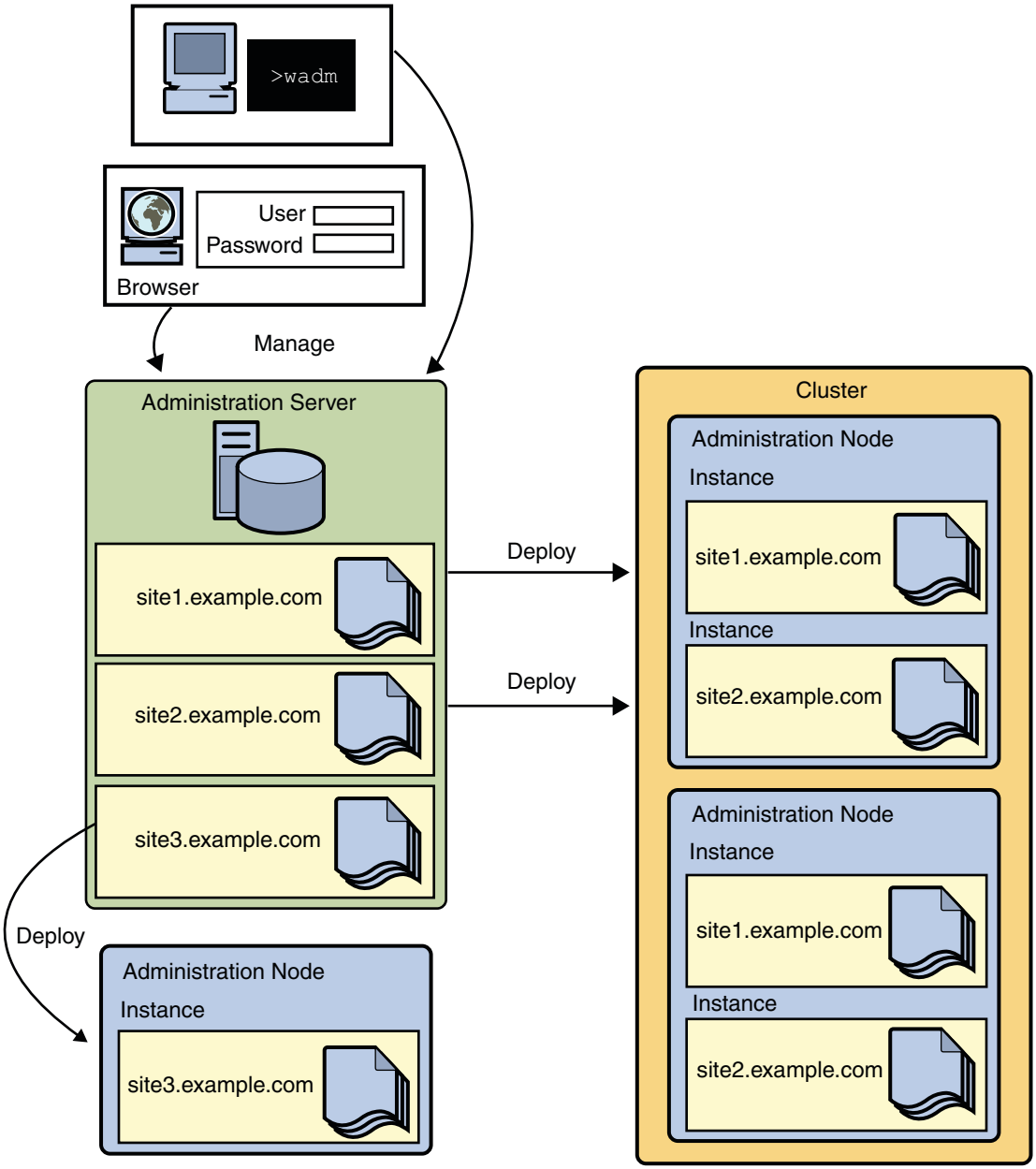
```
<install_dir>/admin-server/config-store/<config_name>/web-app/<virtual_servername>/
```

When you deploy a configuration, the entire web application directory and configuration directory under `config-store` is zipped up and copied to the server instance directory. This file is the `current.zip` file under:

```
<install_dir>/admin-server/config-store/<config_name>
```

Depending on the size of the web application, deploying a selected configuration might take some time to complete.

The following figure shows a schematic diagram of how Configurations are deployed to Administration Nodes:



When you deploy a Configuration to a **Node** (Network resource, such as server or a host), an **Instance** of that Configuration is created. The instance contains log files and other runtime files such as lock databases, caches and temporary files that are required by the instance. You can manage these instances through the CLI or web based administration interface.

Instances can also span across one or more nodes to form a **Cluster**. All nodes that form a cluster must have identical configuration and be homogeneous. They must have the same operating system, be identically configured, and offer the same services.

One node in the server farm has a server running on which the administration application is deployed. This specially configured server is called the **Administration Server** and the administration application that is deployed is the web-based **Administration Console**. You use the Administration Console to control the lifecycle of server instances.

The Administration Server controls the actions of other servers in that node called as **Administration Nodes**. An administration node does not provide a GUI interface. One node in the server farm has the Administration Server installed. All other nodes in the server farm have Administration Nodes installed. An administration Node is registered with an Administration Server upon installation. This action will make the Administration Server aware of that Administration Node.

The Administration server and the administration node always communicate over SSL. The Administration Server and Administration Node authenticate each other by the Administration Server trusting the Administration Node's server certificate and the Administration Node trusting the client certificate presented by the Administration Server. During registration of an Administration Node, the Administration Server will generate a server certificate for that Administration Node, which is then downloaded and installed on the Administration Node. The issuer of the server certificate is also installed on the Administration Node.

◆ ◆ ◆ CHAPTER 2

Configuration, Instances, and Nodes

The previous chapter introduced you to some of the new concepts in Web Server 7.0. The primary task of an administrator is to configure and manage the runtime services of the Server. This chapter describes the different ways by which you can manage Configurations and how you can deploy them to get an instance started on a node.

Overview

Instance refers to the environment of a web server daemon on a given node, including its configuration, log files and other runtime artifacts such as lock databases, caches and temporary files.

A node is a network resource, such as a server or a host. In a typical data center, a network of nodes is called a *server farm*. This section discusses how nodes can be configured using the administration console GUI.

You can deploy one or many instances to a node. The same instance can be deployed to multiple nodes and can be part of different clusters.

For management purposes, an instance can be started, stopped, restarted or dynamically re-configured.

Managing Configurations

Creating a Configuration

In order to start using the web server, you need to create a Configuration.

To create a new Configuration, perform the following tasks:

1. Click the **Configuration tab.** .
2. Click the **New button.**

The wizard guides you through the settings available for creating a Configuration. The following sections provide a description of the fields available in wizard pages:

Step 1 – Set Configuration Information

This wizard page enables you to set the generic information for the new configuration

Set the following parameters in the wizard page:

- **Configuration Name** — Add a new unique name for your configuration.
- **Server Name** — Add a server name for the new configuration. It can be same as the configuration name.
- **Document Root** — Enter a valid document root, wherein all the deployed web applications maintain their directories. The default value is ../docs You can enter the path of any valid directory on the server
- **64 Bit** — Enable/Disable 64–bit support for the web server. The default is *disable*.
- **Server User** — If the server is running on a UNIX-based system, provide a valid user name for the server process. For example, root.

Step 2 — Create a Listener for the Configuration

This wizard page enables you to set the HTTP listener properties for the new Configuration

Set the following parameters in the wizard page:

- **Port** — the Port number where the configuration binds to and listens for requests.
- **IP Address** — the IP address of the host machine. Type * for setting all available IP addresses.

Step 3 — Configure Java, CGI and SHTML

This wizard page enables you to configure properties related to Java/CGI and SHTML.

Set the following parameters in the wizard page:

- **Java** — *Enabled.* By default Java is enabled.**Warning:** Do not disable Java if you need to deploy Java-based web applications using this configuration. Set the home for the Java SE directory. The default value is the directory pointing to the bundled Java SE directory. You can select either the default Java SE directory or specify a new path.

  Note – When the Web Server instance is not serving any Java web applications, you can disable Java using the `disable-java` CLI command or through the administration console. For example, FastCGI and reverse proxy plug-in are non-Java applications. Disabling Java will reduce the memory usage of the Web Server instance. By default, the instance is Java enabled.

  For more information on disabling Java, see the CLI reference `disable-java(1)` and `enable-java(1)`.

- **CGI** — None (Disables CGI support), Enable as File Type (Enables CGI support) and Directory (Specify the URI and path where the CGI documents will be stored).
- **SHTML** — By default SHTML is disabled.

Restore Configuration

The following steps enable you to restore a configuration that was previously deployed.

Note – You can only restore the last seven configurations.

▼ To Restore the Configuration

1. **Click the Configuration tab.**
2. **Click the General sub tab > Restore sub tab.**

   Select the configuration from the configuration backups list.
3. **Click the Restore button.**
4. **Click the Deployment Pending Link on the top right of the Administration Console page to deploy the restored configuration.**

   A pop-up windowappears.

5 **Click the Deploy button.**

Step 4 — Create an Instance

This wizard page enables you to create an instance for the new Configuration.

Set the following parameters in the wizard page:

- Configuration — Name of the new Configuration.
- Select Nodes — Select the nodes for creating an instance of the new configuration. Select nodes from the available list and click the **Add** or **Add All** button to add the nodes.

Note – Using CLI

To create a configuration through CLI, execute the following command:

```
wadm> create-config --doc-root=[DOCROOT] --jdk-home=[JAVAHOME]
--server-user=[SERVERUSER]  [--document-root=serverdocroot] [--platform=32|64]
--http-port=port --server-name=servername CONFIGNAME
```

config1 is the name of the new configuration.

See CLI Reference, `create-config(1)`.

Duplicating a Server Configuration

You can copy a server configuration and create a new configuration. The newly copied configuration is identical to the existing configuration. However, the new configuration will not have any instance even though the configuration from which it has been copied has instances.

To duplicate a configuration, perform the following tasks:

1. Click the **Configuration tab.**.
2. Select the configuration from the list.
3. Click the **Duplicate button.**.
4. In the pop-up window, enter the new configuration name and click OK.

Note – Using CLI

To perform the action through CLI, execute the following command:

```
wadm> copy-config --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 copyconfig1
```

copyconfig1 is the name of the new configuration.

See CLI Reference, `copy-config(1)`.

Deploying the Server Configuration

You need to create a configuration first to deploy on the node.

To deploy an existing configuration, perform the following tasks:

1. Click the **Configurations tab.**
2. Identify the configuration by selecting the configuration checkbox.
3. Click the **Deploy button** .
4. A new window appears, click the **Deploy button** to deploy the configuration.

Deleting the Server Configuration

Note – You cannot delete a configuration if instances of the configuration are deployed to nodes. Even if the instances are deployed and not running, you cannot delete the server configuration. Stop the running instances and undeploy them to delete the configuration.

For deleting a configuration, perform the following tasks:

1. Click the **Configurations tab.**
2. Identify the configuration by selecting the configuration checkbox.
3. Click the **Delete button**.
4. A new window appears, click the **OK button** to delete the configuration.

Pulling Configuration Changes to the Administration Server

Whenever you make manual changes to the configuration, you should replicate the changes back into the Administration server repository as follows:

1. Manually edit the server instance's configuration files as you would do with the earlier version of Web Server (Not recommended).
2. Start the Administration Server.
3. To pull the changes back to the Administration Server repository, execute the following command.

```
wadm> pull-config --user=admin --config=CONFIG_NAME
```

Note – The operation may take some time depending on the configuration.

Note – Always use the Administration Console or the `wadm` CLI to edit the settings. When you invoke `pull-config` only the contents of the `<instance_dir>/config` directory will be pulled into the configuration store from Web Server .

Removing the Administration Node from the Server

▼ To Remove the Administration Node from the Server

1 Click the Nodes tab from the Common Tasks page.

2 Select the Administration Node.

Select the Administration Node from the list

3 Click the Remove button.

Managing Server Instances

Creating a Server Instance

Before creating a new server instance, perform the following checks:

1. Check whether you have created a configuration. Creating a new server instance requires an existing instance configuration to be specified.
2. Check if all the available nodes in the Server Farm already have an instance of the required configuration. You can not create duplicate instances.

Create a new server instance by performing the following tasks:

1. Click the **Configuration tab** and click the configuration link that you want to create an instance.
2. In the New Instance Wizard page, select the configuration for which you need to create an instance. Select the `SMF Service` if you want to create a service for this instance and click the **Next button**.
3. Select the nodes on which instances of the selected configuration [*Step 2*] should exist. Click the **Next button**.
4. View the summary of your selection. Click the **Next button** to view the result of the operation.

Note – Using CLI

To create a server instance, execute the following command:

```
wadm> create-instance --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 serverhost
```

See CLI Reference, `create-instance(1)`.

Starting Server Instances

1. Click the **Nodes tab** to view the list of nodes configured in the server.
2. Select the node by selecting the node name checkbox.
3. Click the **Start Instances** button to open a page window, listing all the instances controlled by that node.
4. Select the instance and click the **Start Instances** button to start the instance.
5. Check if the status of the instance is Running and close the window.

Note – Using CLI

To start a server instance through CLI, execute the following command:

```
wadm> start-instance --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 nodehost1
```

See CLI Reference, `start-instance(1)`.

Stopping Server Instances

1. Click the **Nodes tab** to view the list of nodes configured in the server.
2. Select the node by selecting the node name checkbox
3. Click the **Stop Instances** button to open a page window, listing all the instances controlled by that node.
4. Select the instance and click the **Stop Instances** button to stop the instance.
5. Check if the status of the instance is Not Running and close the window.

Note – Using CLI

To stop a server instance through CLI, execute the following command:

```
wadm> stop-instance --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 nodehost1
```

See CLI Reference, `stop-instance(1)`.

Restarting Server Instances

1. Click the **Nodes tab** to view the list of nodes configured in the server.
2. Select the node by selecting the node name checkbox.
3. Click the **Restart Instances** button to open a page window, listing all the instances controlled by that node.
4. Select the instance and click **Restart Instances** button to restart the instance.
5. Check if the status of the instance is Running and close the window.

Note – Using CLI

```
wadm> restart-instance --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 nodehost1
```

See CLI Reference, `restart-instance(1)`.

Re-Configuring Server Instances

When you make changes to the Configuration, you do not always need to restart the instance. The Administration Server supports re-configuring the server instances to pull changes made to the configuration store. In this Configuration changes are reflected on instances without a server restart. Only dynamically re-configurable changes in the configuration will be affected.

For more information on `reconfig` command see, "Dynamic Reconfiguration" in *Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference*.

Note – Changes in the user, temp-path, log, thread-pool, pkcs11, statistics, CGI, DNS, DNS-cache, file-cache, ACL-cache, SSL-session-cache, access-log-buffer, and JVM (except log-level) settings will not come in to effect after a reconfiguration. Any such changes that require restart will be logged when a reconfiguration is performed. Reconfiguring the file cache requires a server restart.

1. Click the **Nodes tab** to view the list of nodes configured in the server.
2. Select the node by selecting the node name checkbox.
3. Click the **Reconfig Instances** button to open a page window, listing all the instances deployed on that node.
4. Select the instance and click the **Reconfig Instances** button to reconfigure the instance.
5. Check if the status of the instance is Running and close the window.

Note – Using CLI

To re-configure the server instance through CLI, execute the following command:

```
wadm> reconfig-instance --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 serverhost
```

See CLI Reference, `reconfig-instance(1)`.

Deleting Server Instances

Note – The server instance should not be running in order for it to be deleted.

1. Click the **Configuration** tab to view the list of available configurations.
2. Select the configuration from the configurations list.
3. Click the **Instances** sub tab.
4. Select the instance from the list of deployed instances under the **Nodes** section.
5. Select **Delete Instances** from the action drop-down list to delete the selected instance.

Note – Using CLI

To delete a server instance through CLI, execute the following command:

```
wadm> delete-instance --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 serverhost
```

See CLI Reference, `delete-instance(1)`.

Automatically Configuring Instances

Instances can be re-configured or restarted based on scheduled events. You can set a specific time and interval for scheduling automatic instance reconfiguration.

For scheduling events, perform the following tasks:

1. Click the **Configuration** tab and select the configuration.
2. Click the **General sub tab > Scheduled Events sub tab.**

▼ To Add a Scheduled Event

1 Click the configuration tab and then select the configuration from the list that appears.

2 Click General > Scheduled Events sub tab.

3 Click the New button.

4 Configure the following properties:

- **Event**

- *Restart Instances* — This scheduled event will restart all the deployed and running instances for the configuration.
- *Reconfig Instances* — This scheduled event will re—configure all the deployed and running instances for the configuration.
- *Custom Command Line* — Provide the absolute path to a file that will be executed.

- **Schedule**

  The configured time when the event will start. Select the hour and minutes value from the drop-down box.

  - **Every Day** — Starts the event specified every day at the specified time.
  - **Specific Days** — Starts the event specified at specific days.

    1. *Days* — Specify any day from Sunday to Saturday.

    2. *Dates* — Specify any day of the month from 1 to 31 as comma separated entries. E.g. 4,23,9
  - **Specific Months** — Starts the event specified at the specific time and month. Specify month from January to December.
  - **Interval**

    Start the specified event after this time period.

    1. *Every Hours* — Select the number of hours from the drop-down box.

    2. *Every Seconds* — Enter the number of seconds in the text field.

Note – Using CLI

To schedule an event through CLI, execute the following command:

```
wadm> create-event --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --time=10:10 --command=restart
```

See CLI Reference, `create-event(1)`.

▼ To Remove a Scheduled Event

1 **Click the configuration tab and then select the configuration from the list that appears.**

2 **Click General > Scheduled Events sub tab.**

3 **Select the scheduled event and click the Delete button.**

Configuring LDAP Authentication for Administration Server

The Administration Server enables only one predefined administrator login and does not allow user group management. Hence, if multiple users have to login to the Administration Server, LDAP authentication is used. You can login to the Administration Server by using your LDAP userid and password through Administration Console or CLI.

Note – The Administration server by default enables only users belonging to the group `wsadmin` to login. Thus, while enabling LDAP authentication, the administrator can define a list of groups, other than `wsadmin` whose members will be allowed to login.

The LDAP auth-db can also be manually configured to allow the Administration Server to authenticate with LDAP as shown below:

```
<default-auth-db-name>ldap</default-auth-db-name>

<auth-db>
<name>ldap</name>
<url>ldap://ooooxxxxooooo.india.sun.com:389/dc963dindia,dc963dsun,dc963dcom</url>
<property>
<name>bindpw</name>
<value>YWRtaW5hZG1pbg==</value>
<encoded>true</encoded>
</property>
<property>
<name>binddn</name>
<value>cn=Directory Manager</value>
</property>
</auth-db>
```

▼ To Configure LDAP Authentication

1. **Login to Administration Console.**
2. **Click the Nodes tab to view a list of nodes configured in the server.**
3. **Click the Administration Node from the list.**
4. **Select Authentication from Administration Server - General Settings page.**
5. **Select the Use LDAP Authentication button.**

Note – The Use LDAP Authentication is enabled only for Administration Server.

6 Enter LDAP authentication information.

By entering the user groups in the Allowed Groups text field, the administrator enables or disables LDAP authentication to the group.

7 Click the Save button.

Note – Using CLI

- To enable the Administration Server to authenticate against LDAP server, execute the following command.

```
wadm enable-admin-ldap-auth --user=admin --host=serverhost
--password-file=../admin.passwd --port=8989 --ssl=true --no-prompt rcfile=null
--ldap-url=ldap://serverhost.com:3950/dc=xyz,dc=xyz,dc=xyz
--bind-dn=cn="Directory Manager"

wadm enable-admin-ldap-auth --user=admin --host=serverhost
--password-file=../admin.passwd --port=8989 --ssl=true
--ldap-url=ldap://serverhost:port/dc=acme,dc=com
--allow-group="group1,group2,group3"
```

  See CLI Reference, `enable-admin-ldap-auth(1)`

- To disable the Administration authentication to LDAP server execute the following command.

```
wadm disable-admin-ldap-auth --user=admin --host=serverhost
--password-file=../admin.passwd --port=8989 --ssl=true --no-prompt --rcfile=null
```

  See CLI Reference, `disable-admin-ldap-auth(1)`

- To display the Administration LDAP authentication properties execute the following command.

```
wadm get-admin-ldap-auth-prop --user=admin --host=serverhost
--password-file=../admin.passwd --port=8989 --ssl=true --no-prompt rcfile=null

wadm get-admin-ldap-auth-prop --user=admin
--host=serverhost --password-file=../admin.passwd --port=8989 --ssl=true
--no-prompt rcfile=null allow-group
```

  See CLI Reference, `get-admin-ldap-auth-prop(1)`

◆ ◆ ◆ CHAPTER 3

Server Farms and Clusters

The earlier chapters introduced Configuration and how Configuration can be deployed to nodes. In this chapter, you will set up a simple server farm and a cluster.

Cluster Support in Web Server

A cluster is a set of instances, spanning across one or more nodes, all running identical configuration and offering an identical set of runtime services. 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 with which we can manage the servers installed in their respective clusters.

Note – All the instances in a cluster are required to be homogeneous. For example they run on an identical operating system version, use identical patches and service packs, run an identical web server configuration, and offer identical services.

Setting Up a Server Farm

To set up a cluster, you need to first install one administration server and one or more administration nodes. The administration nodes have to be registered to the administration server individually for them to be administered. This action can be performed either during installation of the nodes or after installation through `wadm` CLI.

▼ To Set Up a Server Farm

1 Installing Administration Server and Administration Nodes

Install the administration server. You can install the administration server through Web Server Installer GUI or through the `wadm` CLI.

You can choose the **Express installation** option, which will install an administration server on port 8989. Alternatively, choose **Custom Installation** option for setting your preferences. To install the administration server, choose the option **Install server as Administration Server** in the installer setup screen. You need to specify the SSL port but may or may not specify a non SSL port.

Note – If a non SSL port is specified, an administration node is created in the administration server node and this need not be registered with the administration server explicitly.

To install the administration node , choose **Custom Installation** and then **Install server as administration node**. Specify a port for installation. You do not have an option to select non SSL port because all communication between administration server and administration node is through a secure channel. During installation, you will be asked if you need to register the node with the administration server. If you choose not to register the node during installation, you can use the `wadm` CLI to perform this action.

Note – You cannot install administration nodes through express installation.

---

2 Register the Administration Node with the Administration Server

The administration nodes have to be registered to the administration server for them to be part of the cluster or the server farm. The administration nodes will not start unless they are registered to an administration server. To register the administration node execute the following command through `wadm` CLI.

```
wadm> register-node --user <admin-user> --port <SSL Port> --host <node name>
```

This port is the one specified during the administration server installation. The host is the hostname of the node where the administration server is installed.

This action will register the node to the administration server.

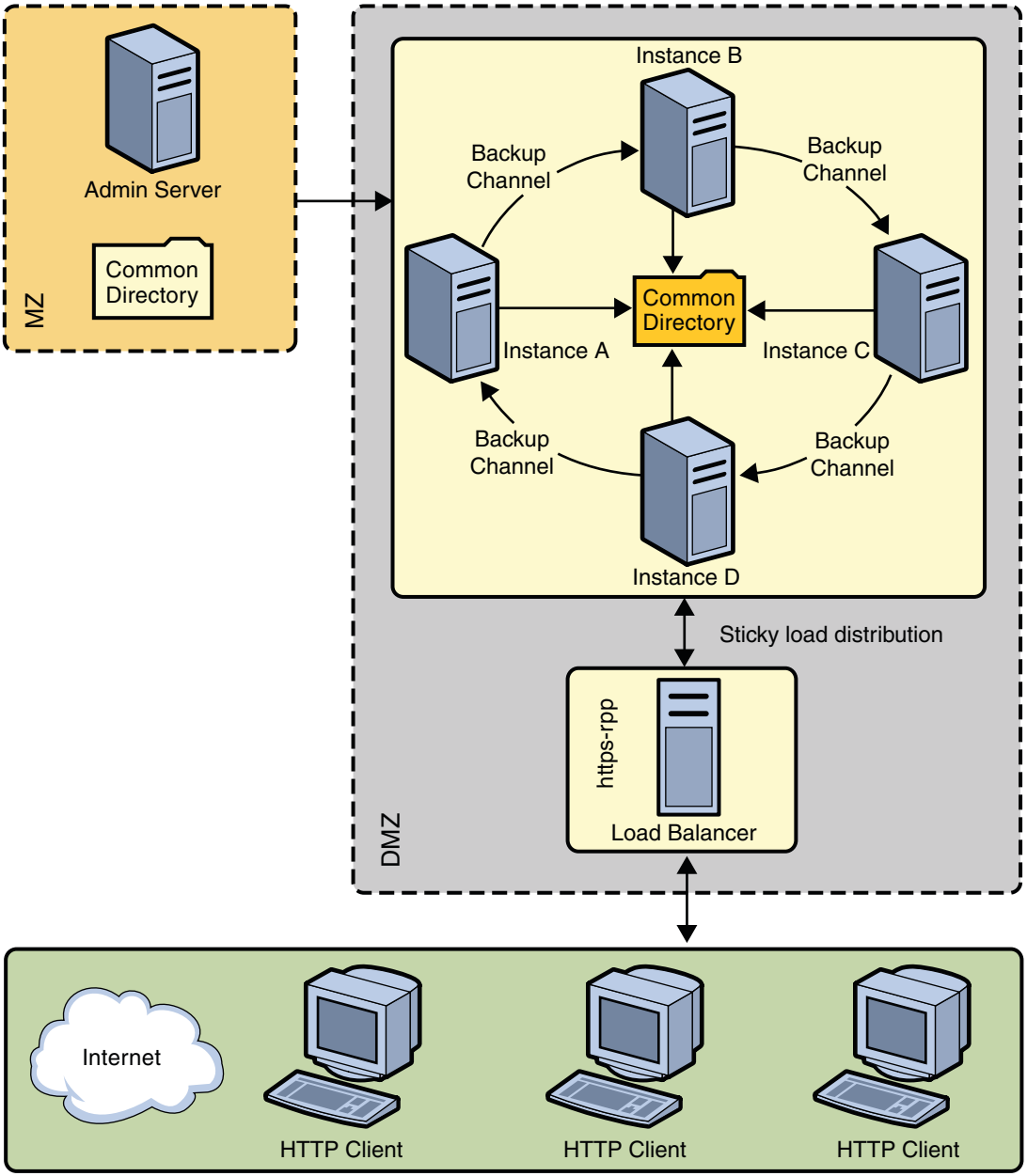
Note – A node can be registered only from the same node . You cannot go to the CLI of the administration server and register any node. Also the registration of node with the administration server can be done in SSL mode only.

To set session replication for the created cluster, see"Configuring Session Replication" on page 196.

Setting Up a Simple Cluster

As part of this example you will set up a cluster with one load balancer, one administration server and four web server instances with session replication enabled. Session replication provides high availability for Java web application sessions. It does so by making copies of sessions resident in memory of one web server instance to another web server instance. So, in normal operational conditions, there are at least 2 copies of every session each residing in a separate JVM and, optimally, on a separate machine.

The following figure depicts a simple cluster:


MZ
Admin Server
Common Directory
DMZ
Instance B
Backup Channel
Backup Channel
Common Directory
Instance A
Instance C
Backup Channel
Backup Channel
Instance D
Sticky load distribution
https-rpp
Load Balancer
Internet
HTTP Client
HTTP Client
HTTP Client

▼ To Configure the Cluster

Before You Begin Identify the following machines:

- MachineA — Has both the load balancer and the administration server.
- MachineB, MachineC, MachineD and MachineE — Has the administration node and the web server instances running.

1 Install Administration Server on MachineA.

See "To Set Up a Server Farm" on page 52 for information on installing an administration server. The typical installation process will also install a web server instance. For this scenario, we will not be using that instance.

2 Install the Administration Node on MachineB, MachineC, MachineD and MachineE.

Install the administration node on all four machines. Register the administration nodes with the administration server.

3 Configure the Web Application.

Enable session replication for the web application. Modify the `WEb-INF/sun-web.xml` file as follows:

```
<session-manager persistence-type="replicated"/>
```

4 Configure the Instances.

- Launch `wadm`.

```
wadm --host MachineA --port 8089
```

- Create a new configuration for the load balancer.

```
wadm> create-config --http-port=8080 --server-name=SampleCluster lb
```

- Set up the reverse proxy (Load balancer).

```
wadm> create-reverse-proxy --config=lb --vs=lb
-uri-prefix=/ --server="http://MachineB:8080,http://MachineC:8080,
ttp://MachineD:8080,http://MachineE:8080"
```

- Create an instance.

```
wadm> create-instance --config=lb MachineA
```

- Deploy the Configuration.

```
wadm> deploy-config lb
wadm> start-instance --config=lb
```

5 Create and Start the Cluster.

Create a new Configuration with four instances.

- Create a new configuration for the cluster.

```
wadm> create-config --http-port=8080 --server-name=SampleCluster clusterOf4
```

- Enable Session Replication.

```
wadm> set-session-replication-prop --config=clusterOf4 enabled=true
```

- Add the web application.

```
wadm> add-webapp --config=clusterOf4 --uri=/simple webapps-simple.war
```

- Create the instances.

```
wadm> create-instance --config=clusterOf4 MachineB MachineC MachineD MachineE
```

- Start the cluster.

```
wadm> start-instance --config=clusterOf4
```

Note – If the host name is not specified for the start-instance command, this action will start instances on all the nodes where the configuration is deployed.

◆ ◆ ◆ CHAPTER 4

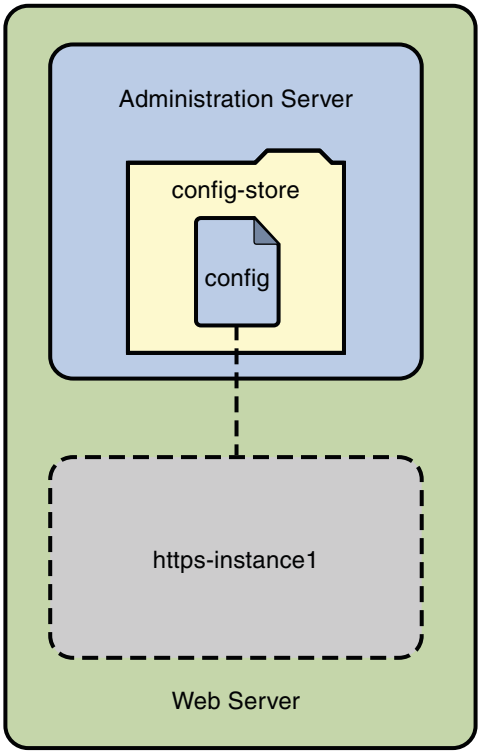
Deployment Scenarios

This chapter describes how to deploy Web Server on a single node and a cluster environment. The following topics are discussed in this chapter:

Deployment Architecture

This section describes the single node deployment architecture.

The following figure represents Web Server in a single node deployment set up.



In the preceding figure, the Web Server deployment set up comprises the following components:

- *Administration Server*- Administration Server is a specially configured web server instance. You can deploy web applications on the administration server.
- *Administration Node*- Administration Node is deployed on a node or a server/host within a server farm and has the ability to communicate with the remote Administration Server. The server configurations available within the Administration Server can be deployed to this node. All the Administration Nodes within the server farm need to be homogeneous. That is, all the nodes must use the same operating system and have the same hardware architecture.
- *Configuration*- A configuration refers to a set of all configurable elements of a Web Server instance, such as web applications, configuration files, and search collection indexes. A configuration can be created, modified, or deleted. Web Server can manage multiple configurations. Instances can be created for a configuration. Deploying a modified configuration updates the instance of that configuration.
- *config-store* This is the file system-based repository where all the configurations are stored.

Caution – Do not edit any file in the `config-store` directory. The files under this directory are created by Web Server for internal use.

If you must manually edit the configuration file in the `config-store` directory, deploy the configuration using the `wadm deploy-config` command.

For more information on using this command, see the *Sun Java System Web Server 7.0 Update 4 CLI Reference Manual*.

- *Instance* - An instance refers to the environment of a web server on a given node, including its configuration, log files, and other runtime artifacts such as lock databases, caches, and temporary files. For management purposes, an instance can be started, stopped, restarted, or dynamically re-configured.

Deployment Overview

You can deploy Web Server on a single node for the following purposes:

- Hosting simple web or CGI applications.
- Developing and testing web applications.

The following flowchart provides the schematic representation of how to deploy Web Server on a single node:

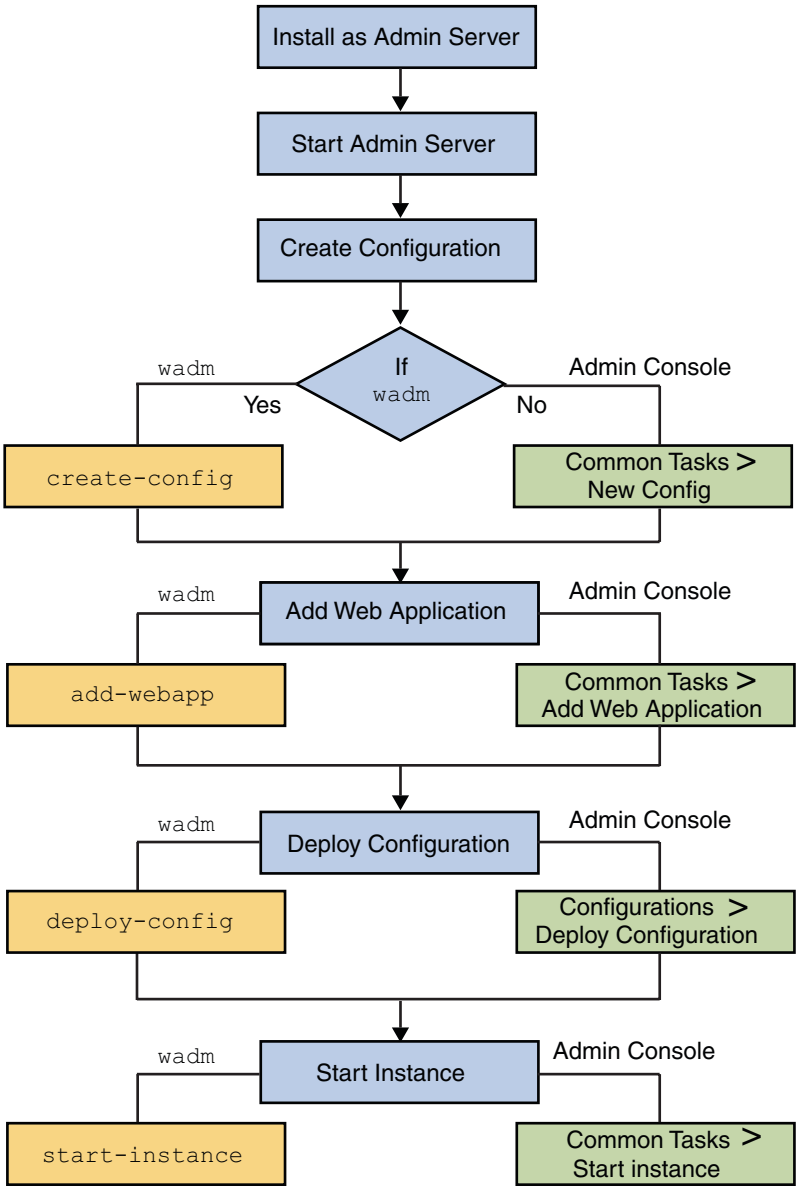


FIGURE 4–1 Flowchart representing the deployment of web server on a single node

The deployment process is described in the following sections:

- "Pre-Deployment Requirements" on page 61
- "Deploying Web Server" on page 61

Pre-Deployment Requirements

To deploy the Web Server on a single node, prepare the system by performing the following tasks:

1. Install Web Server on a node.

   If you choose the Express Installation option while installing Web Server, the following default entities are created:

   - An Administration Server.
   - A default configuration with one HTTP listener and a virtual server are created. The name of the configuration and the virtual server are same as the host name.
   - An instance of the default configuration.

   For information on installing the Web Server, see Chapter 2, "Installing the Web Server," in *Sun Java System Web Server 7.0 Update 4 Installation and Migration Guide*.

   For information on the supported platforms and the system requirements, see "Supported Platforms" in *Sun Java System Web Server 7.0 Update 4 Release Notes*.

2. Start the Administration Server.

   The Administration Server starts running on a specified SSL port.

Deploying Web Server

Use the following procedure to deploy Web Server on a node:

1. You can either use the default configuration or create a new configuration.

   If you are creating a new configuration, specify a unique name for the configuration. The new configuration creates a virtual server and a default HTTP listener.

   Note – If you are using the Administration Console to create a configuration, the wizard prompts you to create a new instance. If you are using the CLI, you must explicitly create an instance of the configuration using the `create-instance` command.

   All the configurations are stored in the `config-store` directory located under <*install_dir*>`/admin-server/` directory.

   Caution – Do not edit any file under the `config-store` directory. The files under this directory are created by Web Server for internal use.

2. Deploy the modified configuration.

Cluster Environment

A *cluster* is a group of multiple server instances spanning across more than one node, all running identical configurations. All instances in a cluster work together to provide high availability, reliability, and scalability.

With load balancing, a cluster offers uninterrupted service and session data persistence by providing failover and session replication.

Hardware and Software Requirements

The use case described in this section, consists of the following entities:

1) Four instances (running on four identical nodes)
2) An Administration Server
3) A reverse proxy for load balancing HTTP requests

To set up a cluster, you need two or more identical nodes with the same operating system version and patches. For example, if you select a machine with Solaris® 9 SPARC® operating system, other machines in the cluster must also have Solaris 9 SPARC installed.

For information on supported platforms and patch requirements, see the *Sun Java System Web Server 7.0 Update 4 Release Notes*.

The following figure describes a clustered environment.

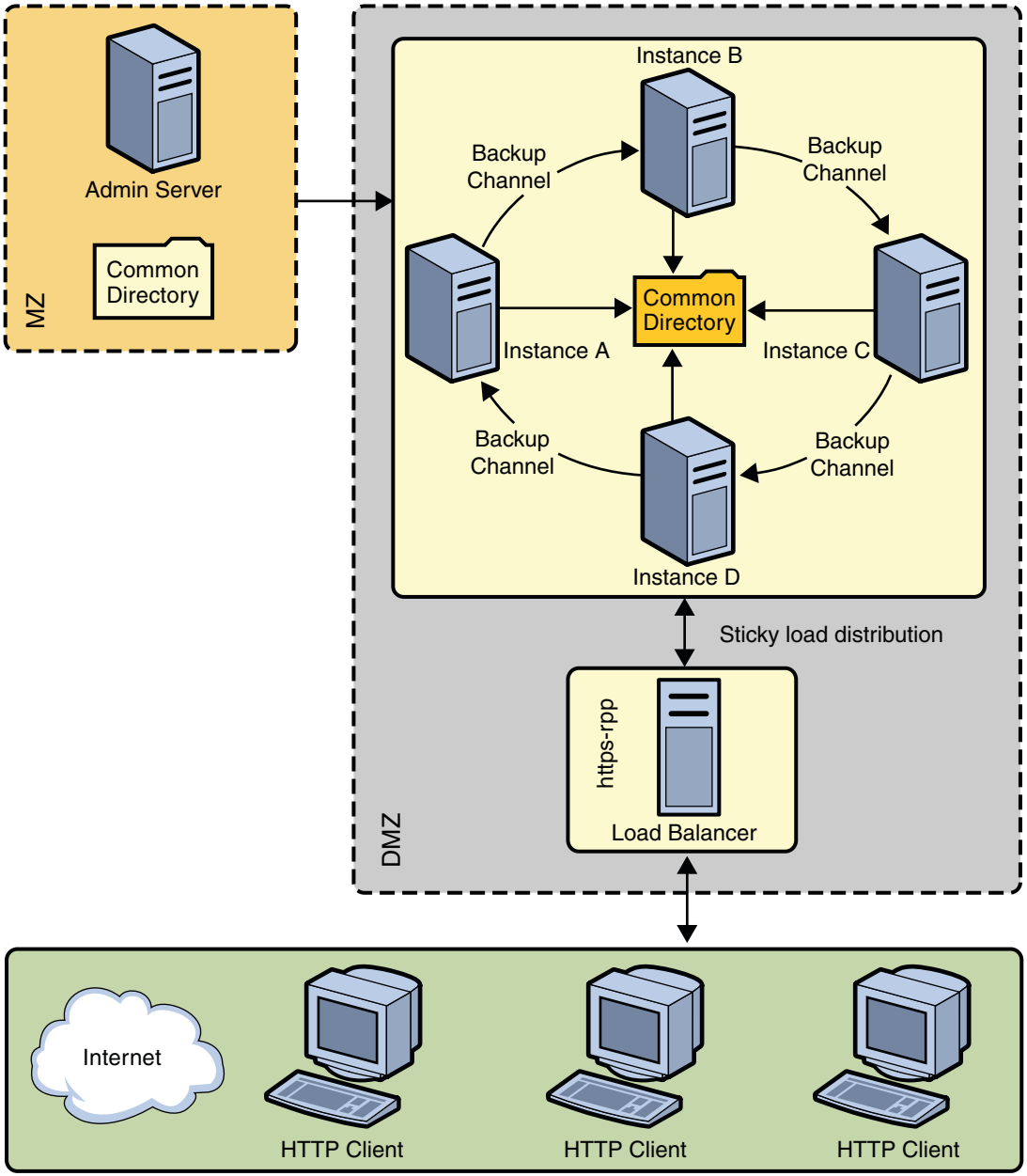


FIGURE 4–2 Cluster Set Up

In the preceding figure, nodes are configured in the De-Militarized Zone (DMZ). The Administration Server is configured behind a firewall, the Militarized Zone, to restrict and

protect the Administration Server against general access. Another node is configured as the Reverse Proxy Server. A reverse proxy server resides inside the DMZ to enhance security.

Note – The Solaris zone feature is supported only on Solaris 10 operating system.

Setting Up a Cluster

This section describes the procedure to set up the cluster and enable reverse proxy to support load-balancing on HTTP requests.

The following flowchart illustrates the procedure to set up a cluster.

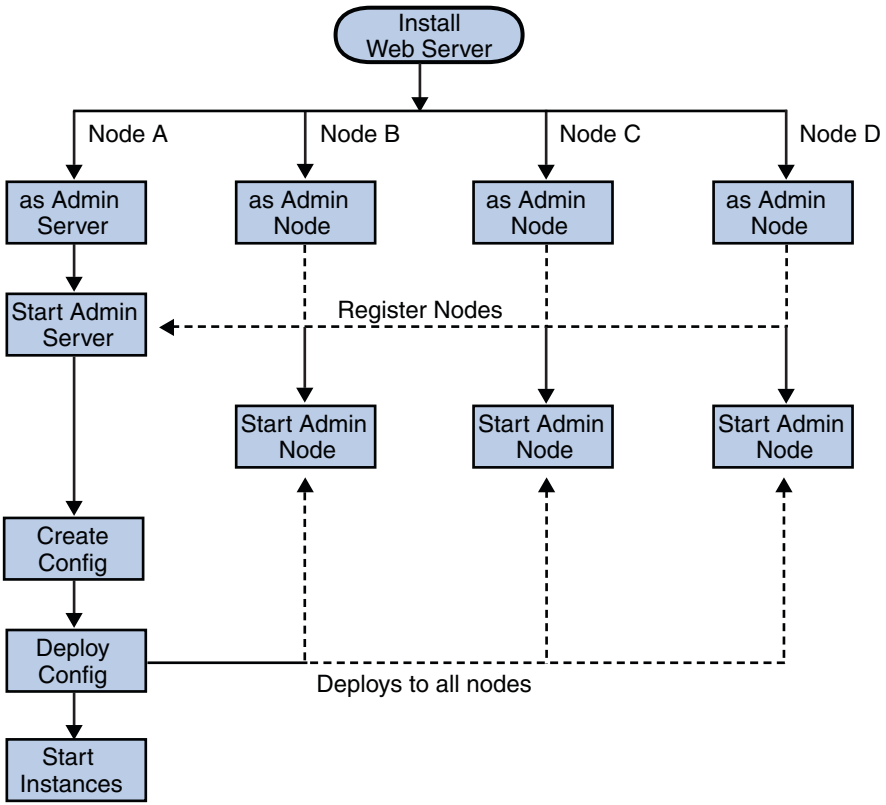


FIGURE 4–3 Flowchart illustrating the cluster set up

1. On one of the nodes, install Web Server that acts as the Administration Server in a cluster.

2. On the other three nodes, install the Web Server. Select the option of installing Web Server as an Administration Node. During the installation, choose the option of registering the node with the server.
3. Make sure the Administration Server is using SSL port for communication, as an Administration Node can be registered with the server only in secure mode.
4. Make sure the system date and time on all the nodes where the Administration Server and the Administration Nodes are installed are the same. The certificate associated with the server is created based on the system date and time of the node where the Administration Server is installed. If the system date of the Administration Node is earlier than the Administration Server, the registration fails as the certificate of the Administration Server will not yet be valid. As a corollary, the certificate may be deemed valid if it is has expired.
5. Start the Administration Server from the *install_dir*`/admin-server/bin/` directory.

   install_dir`/admin-server/bin>./startserv`
6. Start the `wadm` command-line tool from the Administration Node. The `wadm` command-line tool is located in the *install_dir*`/bin` directory.

   install_dir`/bin>./wadm`
7. Register each Administration Node with the Administration Server. Use the `register-node` command to register each node with the server.

   For Example:

   ```
   ./wadm register-node -user=admin --host=abc.sfbay.sun.com --port=8989
   ```

   Where,

`abc.sfbay.sun.com`	is the host name of the Administration Server to which you are registering the Node.
`port`	is the SSL Port number of the Administration Server.
8. You will be prompted to enter the administration password. Enter the administration password of the Administration Server.

   The Administration Server authenticates by the Administration Server trusting the Administration Node's server certificate and the Administration Node trusting the client certificate presented by the Administration Server. During registration of an Administration Node, the Administration Server generates a server certificate for that Administration Node, which is then downloaded and installed on the Administration Node. The issuer of the server certificate is also installed on the Administration Node.

   Note – The registration can be done only over SSL.

For information about registering nodes, see "Registering the Administration Node From the Command-Line" in *Sun Java System Web Server 7.0 Update 4 Installation and Migration Guide*.

9. Start all the Administration Nodes using the `startserv` command from the *install_dir*`/admin-server/bin/` directory.
10. Using the Admin Console or the CLI, create a new configuration in the Administration Server.

    Provide configuration information such as configuration name, HTTP Listener port, and the server name for the new configuration.
11. Create instances of the configuration on all the nodes.
12. Start the instances on all the nodes.

Note – Web Server provides the flexibility to expand or reduce your cluster. You can add or remove instances to the cluster at any point of time.

Configuring Reverse Proxy in Web Server 7.0

The Sun Java System Web Server 7.0 integrates the reverse proxy functionality within the core server.

When web server is configured with reverse proxy functionality, it acts as a proxy for one or more backend servers and serves as a single point of access or gateway in a server farm. In a reverse proxy setup, the web server forwards the HTTP request it received from the browser client to the appropriate backend server. The HTML response from the backend server is sent back to the browser through the web server. Thus, the web server with reverse proxy hides the existence of backend servers.

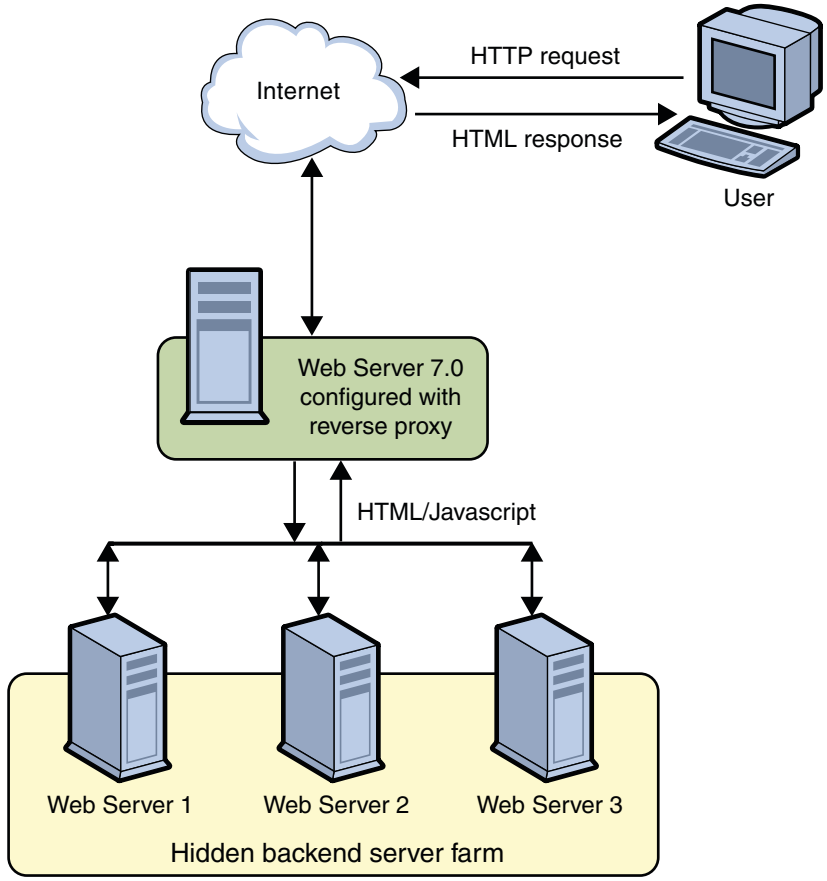


FIGURE 4–4 Reverse Proxy Setup

Web Server 7.0 with reverse proxy functionality acts as a simple software load balancer with the added ability to forward the sticky requests back to the same backend server.

Web server with reverse proxy can serve static content like `gif` and `html` files from its internal cache. At the same time, it functions as a load balancer and processes request for dynamic content like `jsp`, `servlet` or `php` files to the backend server. When web server is deployed in this configuration, disabling the Java web container will significantly reduce the memory footprint of the server. For information about disabling Java web container, see "Tuning Web Container Within Web Server 7.0" in *Sun Java System Web Server 7.0 Update 4 Performance Tuning, Sizing, and Scaling Guide*. See CLI Reference, `disable-java(1)`.

The advantages of reverse proxy within Web Server 7.0 are:

- Conditional processing of request to the backend servers using the integrated regular expression support. For example, you can configure web server to function as reverse proxy only for Servlet and JSP. See "Customizing Reverse Proxy" on page 71.
- Ability to efficiently edit the response received from the backend server using `sed-response` filter before sending the response. For information about `sed-response`, see "sed-response" in *Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference*.
- Ability to dynamically scale the web site by adding more backend servers with less configuration changes and minimal downtime. To add an additional backend server, you should edit the virtual server specific `obj.conf` and run the `reconfig` command. For information about `reconfig` command, see "Dynamic Reconfiguration" in *Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference*.

Note – In a typical deployment, one or more reverse proxies will be deployed between the browsers and the backend servers.

Configuring Reverse Proxy for Load-balancing

Web Server 7.0 provides a sophisticated built-in load balancer, the reverse proxy, which distributes load or request from the client to several backend servers.

Web Server provides GUI and CLI support for configuring the reverse proxy.

▼ Configuring Reverse Proxy Using Administration Console

1. **Install Web Server on the node that you want to use for configuring reverse proxy.**
2. **Create a configuration. For example, `rp`.**
3. **Using the Administration Console, select Configurations > Virtual Servers > Content Handling > Reverse Proxy tab. Click New.**
4. **Specify values for the following parameters:**
   - **URI** — The reverse proxy URI
   - **Server URL** — Comma separated server URLs of all the machines in the cluster separated by comma. If multiple values are given, the server will distribute load among the specified servers.

     The format for entering the server URL is *hostname*:*portnumber*. For example, `http://<content server-hostname>:port`
5. **Click the OK button.**

6 **Click the Deployment Pending link in the top right of the screen to deploy the modified configuration and to apply changes to the configuration.**

7 **Click the Deploy button.**

Deployment successful message appears.

8 **Start all instances of this modified configuration.**

This completes configuring the reverse proxy for load balancing HTTP requests.

Note – To configure a reverse proxy in a cluster environment, issue a wildcard server certificate or the alternate subject names that can be set to the actual origin server host names. The other option of specifying the original server's host names in the subject name field limits the size of the cluster, leading the cluster to fail if another node is added to the cluster.

A wildcard server certificate can be created using the administration interfaces. After creating the server certificate use `certutil` to get the `base64` encoded version of the certificate and install it as a trusted CA certificate on the load balancer configuration.

Type the following command to generate the `base64` encoded certificate `bash$./certutil -L -a -d instancedir/config`. Copy the output of the command and paste it in the install certificate wizard.

▼ Configuring Reverse Proxy Using CLI

Perform the following steps to configure reverse proxy in CLI mode. You will create a configuration `config1` and an instance `rp` as reverse proxy.

1 **Start the Administration Server:**

```
$ <install-dir>/admin-server/bin/startserv
```

2 **Invoke the CLI shell:**

```
<install-dir>/admin-server/bin/wadm -user <username>
```

You can see the `wadm` shell

3 **Create `config1`:**

```
wadm>create-config --http-port=8080 --server-name=config1 --server-user=root
config1
```

4 **Create an instance for the `config1` configuration:**

```
wadm>create-instance --config=config1 <host-name>
```

5 **Add the web application on the created configuration:**

```
wadm>add-webapp --config=config1 -vs=config1 --uri/test <warfile>
```

6 **Deploy the web application.**

```
wadm>deploy-config --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 config1
```

7 **Create a `rp` configuration:**

```
wadm>create-config --http-port=8081 --server-name=rp --server-user=root rp
```

8 **Enable the `rp` configuration to reverse proxy using the following command:**

```
wadm> create-reverse-proxy --user=admin --password-file=admin.pwd
--host=serverhost --config=rp --vs=rp --uri-prefix=//
--server=http://rick.india.sun.com:8080
```

To redirect to a secure site, follow the same step and provide the `https` address for the `--server` option.

See CLI Reference, `create-reverse-proxy(1)`.

9 **Create an instance for the `rp` configuration.**

```
wadm>create-instance --config=rp <host-name>
```

10 **Start the instances:**

```
wadm>start-instance --config=config1 <host-name>
```

```
wadm>start-instance --config=rp <hostname>
```

The web application deployed in `config1` can be viewed through `rp` instance.

```
http://<rp instance hostname>:8081/test
```

See CLI Reference, `list-reverse-proxy-uris(1)`, `set-reverse-proxy-prop(1)`, `get-reverse-proxy-prop(1)`, `forward-reverse-proxy-header(1)`, `block-reverse-proxy-header(1)`, and `list-reverse-proxy-headers(1)`

Modifying Reverse Proxy Parameters

▼ To Modify Reverse Proxy Parameters

1 **Using the Administration Console, select Configurations > Virtual Servers > Content Handling > Reverse Proxy tab.**

2 **Click the URI button.**

You can edit the following parameters:

- **URI** — The reverse proxy URI
- **Server URL** — Comma separated URLs of the remote server. If multiple values are given, the server will distribute load among the specified servers.
- **Sticky Cookie** — Name of a cookie that when present in a response, will cause subsequent requests to stick to that origin server.
- **Sticky URI Parameter** — Name of a URI parameter to inspect for route information. When the URI parameter is present in a request URI and its value contains a colon ':' followed by a route ID, the request will "stick" to the origin server identified by that route ID.
- **Route Header** — Name of the HTTP request header used to communicate route IDs to origin servers.
- **Route Cookie** — Name of the cookie generated by the server when it encounters a sticky cookie in a response. The route cookie stores a route ID that enables the server to direct subsequent requests back to the same origin server.
- **Rewrite Headers** — Comma separated list of HTTP request headers.

Configuring Timeout Parameter in Reverse Proxy

1. Using the Administration Console, select Configurations > Virtual Servers > Content Handling > Reverse Proxy tab.
2. Click the URI button.

   A new window appears.
3. Click the HTTP Client Configuration link.

   You can edit the Idle Timeout parameter. The default value is 300.

Customizing Reverse Proxy

You can configure conditional request processing in reverse proxy by manually editing the virtual server specific `obj.conf` file or through CLI. After the configuration changes are done, it is recommended to deploy the configuration and start the instance so that the changes are implemented.

```
wadm>deploy-config config_name

wadm>start-instance --config config_name hostname
```

See CLI Reference, `deploy-config(1)`, `start-instance(1)`

Note – The appropriate `obj.conf` file used by your virtual server should be modified. It can be `<vs>-obj.conf` or the default `obj.conf`, depending on the configuration.

The following examples discuss some of the possible configurations in Web Server.

- Configuring reverse proxy for all `.jsp`, `.php` requests.
  1. Create a `rp` configuration.

     ```
     wadm>create-config --http-port=8081 --server-name=rp --server-user=root
     rp
     ```

  2. Enable the `rp` configuration to reverse proxy using the following command:

     ```
     wadm> create-reverse-proxy --user=admin --password-file=admin.pwd
     --host=serverhost --config=rp --vs=rp --uri-prefix=//
     --server=http://rick.india.sun.com:8080
     ```

  3. Disable Java for the `rp` configuration.

     ```
     wadm disable-java --user=admin --password-file=admin.pwd
     --host=serverhost --config=rp
     ```

     See CLI Reference, `disable-java(1)`.

  4. Create an instance for the `rp` configuration.

     `wadm>create-instance --config=rp` *<host-name>*

  5. Modify the `<vs>-obj.conf` file, so that the above expression is added to the `NameTrans fn="map"` directive.

     ```
     NameTrans fn="map" from="/" to="http:/" name="custom_reverse_proxy"
     ...
     <Object name ="custom_reverse_proxy">
     Route fn="set-origin-server" server="http://<hostname>:<port>"
     </Object>

     <Object name ppath="http:*"
     Service fn="proxy-retrieve" method="*"
     </Object>
     ```

- Configuring `http-referer header` in reverse proxy.
  1. Create a `rp` configuration.

     ```
     wadm>create-config --http-port=8081 --server-name=rp --server-user=root
     rp
     ```

  2. Enable the `rp` configuration to reverse proxy using the following command:

```
wadm> create-reverse-proxy --user=admin --password-file=admin.pwd
--host=serverhost --config=rp --vs=rp --uri-prefix=/ilearn
--server=http://rick.india.sun.com:8080
```

3. Create an instance for the `rp` configuration.

   `wadm>create-instance --config=rp` *<host-name>*

4. Modify the `<vs>-obj.conf` file, so that the above expression is added to the `NameTrans fn="map"` directive.

```
<Object name="reverse-proxy-/ilearn">
NameTrans fn="set-variable"
$headers{'Referer'}="http://learning.sun.com/TOI/LEARN.html"
Route fn="set-origin-server" server="http://spb-sls-dev.russia.sun.com:7777"
</Object>
```

- Setting up a simple failover scenario for the reverse proxy functionality.

  For example, in a setup there are two reverse proxies which proxy to two separate web servers without load balancing. There is a one to one relationship in a normal scenario. However, if one backend server is down the reverse proxy should send request to the other live web server. Modify the `obj.conf` file as shown below.

```
<Object name="default">

<If $path  =~ '/servlet' or $path =~ '\.jsp'>
<If  not $restarted>
NameTrans fn="map" name="reverse-proxy"  from="/" to="http:"
</If>
<If $restarted>
NameTrans fn="map" name="reverse-proxy-alt"  from="/" to="http:"
</If>
</If>

</Object>

<Object name="reverse-proxy">
Route fn="set-origin-server" server="<back-end-server>"
# If back end server is not available, restart the request
<If $code =~ 504>
Error fn="restart" uri="$uri"
</If>
</Object>

<Object name="reverse-proxy-alt">
Route fn="set-origin-server" server="<alternate-back-end-server>"
</Object>

<Object ppath="http:*">
```

```
Service fn="proxy-retrieve" method="*"
</Object>
```

For every request, the server will first try to reach the first backend server. When this is not available, the request will be sent to the failover server or alternate backend server.

- Setting up a software load balancer to two web server instances that host dynamic content.

  Add server names in the server parameter, separated by a comma (,) and execute the command through CLI.

  ```
  wadm> create-reverse-proxy --user=admin --password-file=admin.pwd
  --host=serverhost --config=rp --vs=rp --uri-prefix=//
  --servers=hostname:port,hostname1:port
  ```

- Configuring `timeout` value for reverse proxy.

  The Web Server 7.0 configured with reverse proxy, returns a gateway timeout error as the backend server takes a long time to respond. You can set the timeout value through CLI as below:

  ```
  wadm> set-reverse-proxy-prop --user=admin --password-file=admin.pwd
  --host=serverhost --config=rp --vs=rp--uri-prefix=//
  --server=http://rick.india.sun.com:8080 timeout=400
  ```

  See CLI reference, `set-reverse-proxy-prop(1)`.

  The default timeout value is 300 seconds. Once the response timeout value is defined, if the connection hangs for more than 400 seconds, the reverse proxy identifies the backend instance offline and closes the connection.

Session Replication

Session replication is a mechanism used to replicate the data stored in a session across different instances. However, the replicated instance must be part of the same cluster. When session replication is enabled in a cluster environment, the entire session data is copied on a replicated instance. However, the session replication operation does not copy the attributes that cannot be serialized in a session and any instance specific data.

Session replication along with load balancing provides good failover capabilities for web applications.

Session Replication and Failover Operation

This section describes the session replication operation in detail.

At the end of a web request, the Web Server determines whether the session data needs to be copied through the session replication configuration that is stored in the server configuration file, the `server.xml`.

Consider a use case of four instances forming a cluster with session replication enabled on the Administration Server.

The session replication process in a Web Server cluster of four instances (A, B, C, and D) running on four nodes is as follows:

- Instance A is backup of D, B is backup of A, C is backup of B, and D is backup of C. This forms a complete backup ring.
- Each instance in the cluster keeps track of a static list of all the instances in the cluster and an active backup instance.
- Depending on the configuration, session data is sent to the backup instance synchronously at the end of each request.

The failover process in a Web Server clustered environmentis as follows:

- The load balancer redirects all incoming web requests destined for instance A to the remaining instances in the cluster and the backup ring is re-configured as follows:
  - D detects that its backup A is down and selects the next instance to A on the ordered list as its new backup instance.
  - B is selected and D establishes a new backup connection to B. B now holds two backups: a read-only backup of A and an active backup of D.

    The backup ring is now complete with B backing up to C, C backing up to D, and D backing up to B.
  - When the failed instance A is made available again, it rejoins the backup ring by sending its designated backup instance B a rejoin message and establishes a backup connection to B.
  - When D detects that Ais online by either receiving a successful ping return from A or by receiving a message from A, D then establishes a backup connection to A and terminates its backup connection to B.

Web Server 7.0 does not support the following features in session replication:

- Recovering the simultaneous failures of two or more instances.
- The interval between two failures must be greater than the time needed for a resurrected instance to fully recover.
- Session backup to more than one instance. In normal operation, there are only two copies of any session: the primary session and a backup session.
- Session persistence: Sessions are only backed up in memory of another instance for the purpose of failover

- Web Server supports session replication for only Java web applications. If you are using non-Java applications such as CGI or PHP, the session data cannot be replicated.

Enabling Session Replication

You can enable session replication in a cluster using either the Admin Console or the CLI. Before you enable session replication, make sure that your browser is cookie enabled.

The `server.xml` file contains the information related to session replication. A sample `server.xml` file with session replication enabled is given below:

```
<cluster>
        <local-host>hostA</local-host>
            <instance>
         <host>hostB</host>
      </instance>
      <instance>
          <host>hostC</host>
      </instance>
       <instance>
          <host>hostD</host>
      </instance>
      <instance>
           <host>hostA</host>
    <session-replication/>
</cluster>
```

If you are using the default values for the following elements, the entry for these elements will not be available in the `server.xml` configuration file:

`Port number` (default is 1099)
`Protocol` (default is `jrmp`)
`Encrypted` (default is false)
`Getattribute Triggers Replication` (default is true)
`Replica Discovery MaxHops` (default is –1)
`Startup Discovery Timeout` (default is 0. Relies on Java API to get system timimg. In non Unix based operating systems, it may not be accurate.)
`Cookie Name` (default is `CLUSTERSESSIONLOCATOR`)

For more information about these session replication properties, see the *Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference*.

Configuring a Web Application for Session Replication

To enable the server to replicate the session, the web application must also be enabled for session replication.

1. To enable session replication for a web application, modify the `sun-web.xml` configuration file located in the *<web-application>*`/WEB-INF` directory.

   The modification needed in the `sunweb.xml` is as follows:

   Change the element `<session-manager/>` to
   `<session-manager persistence-type="replicated">`

   The sample `sun-web.xml` file with session replication enabled is given below:

```
<sun-web-app>
        <session-config>
     <session-manager persistence-type="replicated">
     </session-manager>
  </session-config>
</sun-web-app>
```

2. After modifying the `sunweb.xml` file, either rebuild the web application or re-jar the application to create a web application archive (a war file).
3. Restart all the instances to make the web application available on all the instances.
4. The web application is accessible from all the nodes in the cluster. To access the web application, in a browser, type the following:

   `http://`*webserver-name*`/`*webapplication-name*`/`

Note – A directory that is accessible to all nodes is the best way to store the applications for deployment. This directory, however, need not be accessible to the Administration Server. It is recommended to make directory-based deployments of web applications that are more than 1 MB in size.

To create search collections, ensure that the search collection resides in a common directory that is accessible to all the nodes.

Monitoring a Cluster

The Administration Server can monitor all the instances in a cluster. The monitoring feature in Web Server provides information on the state of runtime components and processes that can be used to:

- Identify performance bottlenecks
- Tune the system for optimal performance

- Aid capacity planning
- Predict failures
- Perform root cause analysis in case of failures

Solaris Zones

Solaris Zones are an application and resource management feature of Solaris 10. A zone environment typically consists of resources such as process management, memory, network configuration, file systems, package registries, user accounts, shared libraries, and in some cases, installed applications. Zones provide a means of creating virtualized operating system environments within an instance of Solaris, allowing one or more processes to run in isolation from other activity on the system. They also provide an abstraction layer that separates applications from physical attributes of the machine on which they are deployed, such as physical device paths and network interface names, and network routing tables. This isolation prevents processes running within a given zone from monitoring or affecting processes running in other zones, regardless of user ID and other credential information.

A zone is a *sandbox* within which one or more applications can run without affecting or interacting with the rest of the system.

For detailed information about Solaris Zones, see the *System Administration Guide — Solaris Containers-Resource Management and Solaris Zones* at `http://docs.sun.com/app/docs/doc/817-1592`.

◆ ◆ ◆ CHAPTER 5

Using Virtual Servers

Overview of Virtual Servers

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.

All virtual servers have an HTTP Listener specified. When a new request comes in, the Server determines which virtual server to send it to based on the configured HTTP Listener.

Use Cases

Server instances can have any number of HTTP Listeners, both secure and non-secure. You can have both IP-address-based and URL-host-based virtual servers.

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 Web Server .

Default Configuration

After a new installation of the Web Server , you have one server instance. This server instance has just one HTTP Listener listening on the port you selected during 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 HTTP Listener is configured to listen on port 80 on any address to which that machine is configured ("ANY:80" or "0.0.0.0:80").

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 HTTP Listeners.

Secure Server

See "Configuring SSL for the Server" on page 103.

Intranet Hosting

A more complex configuration of the 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 can 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 HTTP Listeners. 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.
- URL-host-based virtual servers do not work with legacy HTTP clients.

You can also set up the IP-address-based configuration with one HTTP Listener per address.

Intranet hosting using separate HTTP Listeners

The advantages of 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 do not 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 HTTP Listeners may give you a minimal performance gain because the server does not have to find out the address where the request arrived. However, using multiple HTTP Listeners also results in additional overhead (memory and scheduling) because of the additional acceptor threads.

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. For example, you can have one configuration that enables only static content, and another one that enables static content and CGIs.

Managing Virtual Servers

Adding a Virtual Server

▼ To Add a Virtual Server

Before You Begin

- Create and identify a configuration for which you need to create a virtual server.
- Create a HTTP listener.
- Identify the host(s) for the new virtual server.

1. **Select the configuration from which you will need to add a virtual server. You can select the configuration from the list of configuration shown in the Configurations tab.**
2. **Click on the Virtual Servers tab > New button.**

   A pop-up wizard page appears to guide you through the Virtual Server configuration process.
3. **From the wizard page, perform the following tasks:**
   - Enter the new virtual server information:
     a. Enter a name to identify the new virtual server. The name can be alphanumeric and can also include period (.), dash (-) and underscore (_) characters.
     b. *(Optional)* Enter a list of hosts to be added to the new virtual server.
     c. *(Optional)* Enter the document root for the virtual server.
   - Select an HTTP listener for the newly configured virtual server. You can either choose an existing HTTP listener or create a new HTTP listener.
4. **The wizard summary page appears. To change the configuration, go back to the previous pages by clicking Previous. Click Finish to complete the new virtual server configuration process.**
5. **The Results page appears. If you see any error, Configure the virtual server again by going back to the previous pages in the wizard.**

Note – Using CLI

To add a virtual server through CLI, execute the following command:

```
wadm> create-virtual-server --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --document-root=../docs config1_vs_1
```

See CLI Reference, `create-virtual-server(1)`.

Configuring a Virtual Server

To configure virtual server's general settings, perform the following task:

To Configure a Virtual Server

1. **Select the configuration from the configuration list. Click Configurations tab to get the list of available configurations.**
2. **Select the virtual server from the virtual server list. Click Virtual Servers tab to get the available virtual servers for the selected configuration.**
3. **Click General tab. Configure the following settings.**
   - **Enabled** — Whether the virtual server is enabled at runtime.
   - **Document Root** — The document root path for the virtual server, where the virtual server's data will be stored. This includes exploded web application directories and log files.
   - **Hosts** — You can enter more than one URL host, separated by commas.

Duplicating a Virtual Server

To duplicate a virtual server, perform the following task:

To Duplicate a Virtual Server

1. **Select the configuration from the configuration list. Click the Configurations tab to get the list of available configurations.**
2. **Select the virtual server from the virtual server list. Click the Virtual Servers tab to get the available virtual servers for the selected configuration.**
3. **Click, Copy and provide a name for the new virtual server.**

Note – Using CLI

To duplicate a virtual server through CLI, execute the following command:

```
wadm> copy-virtual-server --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 copiedVs
```

`copiedVS` is the name of the new virtual server.

See CLI Reference, `copy-virtual-server(1)`.

Configuring HTTP Listeners

The Server accepts the HTTP requests via an HTTP Listener before forwarding the request to the configured Virtual Server. This page enables you to add and configure HTTP Listeners.

HTTP Listeners must have a unique combination of port number and IP address. You can use either IPv4 or IPv6 addresses. Setting the IP address to "*" creates an HTTP Listener that listens on all IP addresses on that port.

Creating a HTTP Listener

You can create a new HTTP Listener for a Virtual Server for processing incoming HTTP requests performing the following steps:

1. Click the **Virtual Servers** tab under Configurations tab.
2. Click the **HTTP Listeners** sub tab to view the list of configured HTTP Listeners.
3. Click the **New** button to open a wizard page that creates a new HTTP Listener.

Provide the following information in the wizard page.

- **Name** — Name for the new HTTP Listener.
- **Port** — Port for the HTTP Listener to bind and listen for incoming HTTP requests.
- **IP Address** — A Valid IPv4 or IPv6 address. "*" implies that the HTTP Listener will listen to all IP addressed for the configured port.
- **Server Name** — Enter the server name E.g. *sales.mycomp.com*
- **Default Virtual Server** — Select the virtual server from the drop-down list. This action will associate the new HTTP Listener for the selected virtual server.
- **Description (Optional)** — Enter a short description for your HTTP Listener.

Note – Using CLI

For creating a HTTP Listener through CLI, execute the following command.

```
wadm> create-http-listener --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --listener-port=18003 --config=config1 --server-name=config1.com
--default-virtual-server-name=config1_vs_1 config1_ls_1
```

See CLI Reference, `create-http-listener(1)`.

Configuring Your HTTP Listener

You can edit the existing HTTP Listener settings by performing the following tasks:

1. Click the **Virtual Servers** tab under a Server Configuration, for editing an existing HTTP Listener setting.
2. Click the **HTTP Listeners** sub tab to view the list of configured HTTP Listeners.
3. Under the **Listener Name** table column, click the HTTP Listener you need in order to edit its setting.

You can edit both the general and security related settings for the HTTP Listener.

Modifying HTTP Listener Parameters

Click the **General tab** to edit basic and advanced HTTP Listener settings. Configure the following options:

- **Name** — Name for the new HTTP Listener
- **Port** — Port for the HTTP Listener to bind and listen for incoming HTTP requests.
- **IP Address** — A Valid IPv4 or IPv6 address. "*" implies that the HTTP Listener will listen to all IP addressed for the configured port.
- **Server Name** — Enter the server name E.g. *sales.mycomp.com*

Select the **Configure Advanced Settings option** under the **Advanced section** to edit HTTP Listener advanced settings. Configure the following options:

- **Acceptor Threads** — Number of threads dedicated to accept connections received by this listener. Accepted values are 1 to 128.
- **Protocol Family** — The protocol used by the listener. Do not modify this value. Default is HTTP.
- **Listen Queue Size** — Maximum size of the operating system listen queue backlog.
- **Receive Buffer Size** — Size (in bytes) of the operating system socket receive buffer.

- **Send Buffer Size** — Size (in bytes) of the operating system socket send buffer.
- **Blocking I/O** — Determines if the HTTP listener socket is in the blocking mode. Disabled by default.

◆ ◆ ◆ CHAPTER 6

Certificates and Keys

This chapter describes the use of certificates and keys authentication to secure the server. It describes how to activate various security features designed to safeguard data, keep out intruders, and allow access.

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.

Using Certificates for 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.

Certificates or digital certificates are collections of data that uniquely identify or verify an individual, company, or other entity on the Internet. Certificates also enable secure, confidential communication between two entities. Personal certificates are used by individuals, whereas server certificates are used to establish secure sessions between the server and clients through secure sockets layer (SSL) technology.

A certificate is like a passport; it identifies the holder and provides other important information. Certificates are verified, issued and digitally signed by a trusted third party called Certification Authority (CA). Once a CA has signed a certificate, the holder can present it as proof of identity to establish encrypted, confidential communication. 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.

Certificates are based on public key cryptography, which uses a pair of digital keys (very long numbers) to encrypt (encode) information, so that it can be read only by its intended recipient. The recipient then decrypts (decodes) the information to read it.

A key pair contains a public key and a private key. The owner distributes the public key and makes it available to anyone. But the owner never distributes the private key; it is always kept secret. Because the keys are mathematically related, data encrypted with one key can be decrypted only with the other key in the pair. Most importantly, a certificate binds the owner's public key to the owner's identity.

In addition to the public key, a certificate typically includes information such as:

- The name of the holder and other identification, like the URL of the Web Server using the certificate, or an individual's email address
- The name of the CA that issued the certificate
- The "digital signature" of the issuing CA
- The validity period (the certificate remains valid only within this period and not before or after this period)

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. SSL enabled servers must have a certificate and clients may optionally have a certificate.

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.

Certificate Chain

Digital certificates are verified using a chain of trust. The trust anchor for digital certificate is the root Certificate Authority (CA). Web browsers are preconfigured with a set of root CA certificates that the browser automatically trusts. Any certificate from elsewhere must come with a certificate chain to verify its validity.

A certificate chain is a sequence of certificates, where each certificate in the chain is signed by the subsequent certificate, eventually resulting in a tree structure. A certificate chain thus traces the path of a certificate from a branch to the root in the hierarchy. The root certificate is a self-signed, topmost certificate of the tree and is generated first. A self-signed certificate is one for which the issuer (signer) is the same as the subject (the entity whose public key is being authenticated by the certificate). The certificates that are directly subordinate to the root certificate have CA certificates that are signed by the root certificate. All certificates below the root certificate thus inherit the trustworthiness of the root certificate.

A certificate chain has the following components:

- A root CA certificate
- One or more intermediate certificates
- Client/server certificate signed by the intermediate CA certificate

In a certificate chain:

- Each certificate is followed by the certificate of its issuer. The certificate contains the distinguished name of the certificate's issuer and is same as the subject name of the next certificate in the certificate chain.
- Each certificate is signed with a private key of its issuer. The signature can be verified with the public key in the issuer's certificate, which is the next certificate in the certificate chain.

Verifying a certificate chain is a process of ensuring that a specific chain is valid, correctly signed, and trustworthy. The purpose of certificate chain is to establish a chain of trust from a subordinate certificate to a trusted root CA certificate. The root CA certificate vouches for the identity in the branch certificate by signing it. If the root CA is the one you trust, it implies that you can trust the certificate of its branches.

During a certificate chain verification, the authentication will fail when:

- The root CA is not trusted
- An invalid signature is found
- The certificate validity dates are expired

Certificate Key Types

In addition to RSA keys, Web Server introduces support for Elliptic Curve Cryptography (ECC).

ECC is emerging as an attractive public-key cryptosystem because compared to traditional cryptosystem like RSA, ECC offers equivalent security with smaller key sizes, which results in faster computations, lower power consumption, and memory and bandwidth savings. Elliptic Curve Cryptography (ECC) has been endorsed by the US government.

It is now possible to select whether you want to generate a certificate request or a self-signed certificate using RSA keys or ECC keys.

For RSA keys different key sizes can be provided (bigger key sizes means better encryption. Default key size is 1024). For ECC keys you should choose the curve the keypair will be generated on. A number of curves have been named by various organizations (ANSI X9.62, NIST, SECG) and Web Server supports all the ones currently specified.

If you intend to request a certificate from a CA (instead of using a self-signed certificate) be sure to contact your preferred CA first to obtain their latest information regarding ECC usage. Ask if they recommend a particular ECC curve for your use case(s). If you do not have guidance on curve selection from your CA or from your organizations internal policies, here are some recommendations. Keep in mind that since ECC is an emerging technology it is possible that curve recommendations for particular use cases may have changed from the time this document was written.

Some supported ECC Curves are listed below:

prime256v1
secp256r1
nistp256
secp256k1
secp384r1
nistp384
secp521r1
nistp521
sect163k1
nistk163
sect163r1
sect163r2
nistb163
sect193r1
sect193r2
sect233k1
nistk233k1

nistk233
sect233r1
nistb233
sect239k1
sect283k1
nistk283
sect283r1
nistb283
sect409k1
nistk409
sect571k1
nistk571
sect571r1
nistb571
secp160k1
secp160r1
secp160r2
secp192k1
secp192r1
nistp192
secp224k1
secp224r1
nistp224
prime192v1

Creating a Self-Signed Certificate

You can generate a self-signed certificate if you do not need your certificate to be signed by a CA, or if you wish to test your new SSL implementation while the CA is in the process of signing your certificate. This temporary certificate will generate an error in the client browser to the effect that the signing certificate authority is unknown and not trusted.

To create a self-signed certificate through CLI, execute the following command.

```
wadm> create-selfsigned-cert --user=admin --port=8989 --password-file=admin.pwd
--config=config1 --token=internal --org-unit=org1 --locality=XYZ --state=DEF
--validity=10 --org=sun --country=ABC --server-name=serverhost --nickname=cert1
```

See CLI Reference, `create-selfsigned-cert(1)`.

Importing Self-signed Certificate to IE Browser

The Web Server installer should import the admin self-signed certificate into the IE certificate tab. When the Admin console is accessed using a browser, a pop-up window (in the case of IE6

and Mozilla/Firefox) or a warning page (IE7) may appear stating that the certificate is not issued by a trusted certificate authority. This is because the administration server uses a self-signed certificate. To proceed to the Administration GUI login page, do the following:

- OnMozilla/Firefox, click the**OK** button in the pop-up window.
- On Internet Explorer 6, click the **Yes** button in the pop-up window.
- On Internet Explorer 7, click the **"Continue to this web site"** link in the page.

These procedures will accept the certificate temporarily for that browser session. To accept the certificate permanently, follow the steps below:

- On Firefox/Mozilla:

  Select the **Accept this certificate permanently** radio button in the pop-up window and click **OK**.
- On Internet Explorer 6.0:
  1. Click the **View Certificate** button in the pop-up window.

     Another pop-up window appears
  2. Click the **Certification Path** tab and select the `admin-ca-cert`.
  3. Click the **View Certificate** button and then click the **Install Certificate...** button. This action invokes the certificate import wizard using which you can import the admin CA certificate into the trusted root certificate database.
- **On Internet Explorer 7:**
  1. Click the **Continue to this website** link on the warning page. The login page is displayed.
  2. Click the **Certificate Error** link located next to the address bar. A warning window is displayed. Click the **View certificates** link.
  3. Follow the steps 1 to 3 as described in the section On Internet Explorer 6 to import the admin CA certificate into the trusted root certificate database.

Managing Certificates

Requesting a Certificate

You can request a certificate and submit it to a 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 information they require. You can also create a self-signed certificate for the server. Self-signed certificates are not suitable for Internet-facing deployments but can be very useful for development and testing because they allow you to set up test servers without CA involvement.

As mentioned above, a certificate includes the public key of the entity (the web server in this case). A public key is generated based on a particular algorithm (the algorithm type is also encoded in the certificate). The next section provides background on the algorithm types supported by the Web Server for its keys.

▼ To Request a Certificate

1. **Click Server Certificates tab > Request button.**
2. **Select a configuration from the configuration list for which you need to install the certificate.**
3. **Select the token (Cryptographic Device), which contains the keys.**
4. **If your key is stored in the local key database maintained by the server, choose internal. If your key is stored in a Smart Card or other external device or engine, choose the name of the external token from the drop-down list box. Enter the password for the selected token.**
5. **Enter Details.**

   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:

   - **Server 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 do not match, a client is notified that the certificate name doesn't match the site name, creating doubt about the authenticity of your certificate.

     You can also enter wildcard and regular expressions in this field if you are requesting a certificate from an internal CA. Most vendors will not approve a certificate request with a wildcard or regular expression entered for common name.
   - **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, for the organization.
- **State or Province** is optional.
- **Country** is a 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 forms the subject of the certificate.

6 Choose Certificate Options

You are required to provide the key information. For key type, you can choose RSA or ECC. If the key type is RSA, the key size can be 1024, 2048 or 4098. If your key type is ECC you will also need to select a curve. Keep in mind that generating a new key pair takes time. The longer the key length the longer the time the wizard takes to generate it.

Caution – Be sure to select a key type that the CA (to which you will later submit the request for signing) can support.

7 Select Certificate Type.

Select the Certificate Signing Authority (CSA) for the certificate (self-signed or CA signed). If you are selecting a self-signed certificate, you can also associate an HTTP Listener for the certificate. You can also perform this action later.

8 Generate Request.

The generated certificate request will be available in ASCII format in case of CA signed certificate. In case of self-signed certificate, it will be installed directly. If the type is self-signed, provide values for nickname, validity (Months) and the HTTP Listener name for handling secure requests.

9 View Results.

This page provides you with the summary of selected options. Click Finish to complete the request generation.

Note – Using CLI

To request a certificate through CLI, execute the following command.

```
wadm> create-cert-request --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --server-name=servername.org
--org=sun --country=ABC --state=DEF --locality=XYZ --token=internal
```

See CLI Reference, `create-cert-request(1)`.

Note – To create a self-signed certificate through CLI, see "Creating a Self-Signed Certificate" on page 91.

Configuring Solaris Cryptographic Framework

This section describes how to configure Solaris cryptographic for use with Web Server.

▼ To Configure Solaris Cryptographic

1 **Remove the `./sunw` directory from your machine using the following command:**

```
%rm -rf $HOME/.sunw
```

2 **Set a new pin using the following command:**

```
% pktool setpin Enter new PIN:<type the pin here>

Re-enter new PIN:<retype the pin again>
```

3 **Disable the mechanisms in the `pkcs11_kernel.so` and `pkcs11_softtoken.so` files using the following command:**

```
#cryptoadm disable provider=/usr/lib/security/$ISA/pkcs11_kernel.so
mechanism=CKM_SSL3_PRE_MASTER_KEY_GEN,CKM_SSL3_MASTER_KEY_DERIVE,CKM_SSL3_KEY
AND_MAC_DERIVE,CKM_SSL3_MASTER_KEY_DERIVE_DH,CKM_SSL3_MD5_MAC,CKM_SSL3_SHA1_MAC

#cryptoadm disable provider=/usr/lib/security/$ISA/pkcs11_softtoken.so
mechanism=CKM_SSL3_PRE_MASTER_KEY_GEN,CKM_SSL3_MASTER_KEY_DERIVE,CKM_SSL3_KEY
AND_MAC_DERIVE,CKM_SSL3_MASTER_KEY_DERIVE_DH,CKM_SSL3_MD5_MAC,CKM_SSL3_SHA1_MAC
```

Note – Ensure to disable mechanisms in `pkcs11_softtoken_extra.so` file, if it is used.

▼ To Register PKCS#11 Library File

1 **Type the following command to add the Solaris crypto framework to network security services (NSS) in the config directory**

```
$ cd <install-dir>/<instance-dir>/lib/modutil -dbdir
<install-dir>/<instance-dir>/config -nocertdb -add "scf" -libfile
/usr/lib/libpkcs11.so -mechanisms RSA
```

2 **Verify the registration using the following command:**

```
$cd <install-dir>/<instance-dir>/lib/modutil -dbdir
<install-dir>/<instance-dir>/config -nocertdb -list
```

```
Listing of PKCS #11 Modules
1. NSS Internal PKCS #11 Module
     slots: 2 slots attached
         status: loaded

          slot: NSS Internal Cryptographic Services
         token: NSS Generic Crypto Services

          slot: NSS User Private Key and Certificate Services
         token: NSS Certificate DB

   2. scf
         library name: /usr/lib/libpkcs11.so
          slots: 1 slot attached
         status: loaded

          slot: Sun Crypto Softtoken
         token: Sun Software PKCS#11 softtoken

   3. Root Certs
         library name: libnssckbi.so
          slots: There are no slots attached to this module
         status: Not loaded
```

For more information on creating server certificates, see "Requesting a Certificate" on page 93

If certificates exists in the NSS database, you can export or import the certificates using the following `pk12util` command:

```
$pk12util -o server.pk12 -d . -n <server-cert>
```

```
$pk12util -i server.pk12 -d . -h "Sun Software PKCS#11 softtoken"
```

Note – By default, `certutil/pk12util` searches the databases for `cert8.db` and `key3.db`. Add `-P` as the prefix for the Web Server, which uses the alternate names `https-instance-hostname-cert8.db` and `https-instance-hostname-key3.db`.

▼ To Enable and Bypass PKCS#11 Tokens

1 **From the home page, click the Configurations tab.**

2 **In the Configuration page, click the configuration that you want to enable the PKCS#11 and Allow Bypass option.**

3. **Click the Certificates tab.**
4. **Click the PKCS#11 Tokens sub tab.**
5. **In General Settings, select the check boxes to enable PKCS#11 and Allow Bypass.**
6. **Click the Save button.**

   See CLI reference, `set-pkcs11-prop(1)`.

Creating a Self-Signed Certificate Using CLI and Enabling SSL

Start the `wadm` from the installation directory and perform the following steps:

```
$wadm --user=admin
Please enter admin-user-password>enter the administration serverpassword

$wadm>list-tokens --config=test.sun.com

internal
Sun Software PKCS#11 softtoken

$wadm>create-selfsigned-cert --config=test.sun.com --server-name=test.sun.com --nickname=MyCert
--token="Sun Software PKCS#11 softtoken"
Please enter token-pin>enter the password

CLI201 Command 'create-selfsigned-cert' ran successfully

$wadm>set-ssl-prop --config=test.sun.com --http-listener=http-listener-1 enabled=true
server-cert-nickname="Sun Software PKCS#11 softtoken:MyCert"
CLI201 Command 'set-ssl-prop' ran successfully

$wadm>deploy-config test.sun.com
CLI201 Command 'deploy-config' ran success
```

Start the Administration Server.

```
$ cd <install-dir>/<instance-dir>/bin/startserv
Sun Java System Web Server 7.0 Update 3

Please enter the PIN for the "Sun Software PKCS#11 softtoken" token:enter the password
info: HTTP3072: http-listener-1: https://test.sun.com:2222 ready to accept requests
info: CORE3274: successful server startup
```

Installing a Certificate

After obtaining the certificate from a CA, you can install the certificate for a configuration using the Administration Console.

▼ To Install a Certificate

1. **Click the Server Certificates tab > Install button.**

2. **Select Configuration.**

   Select a configuration from the configuration list for which you need to install the certificate.

3. **Select Tokens.**

   Select the token (Cryptographic Device) which contains the keys. If your key is stored in the local key database maintained by the server, choose internal. If your key is stored in a Smart Card or other external device or engine, choose the name of the external token from the drop-down list box. Enter the password for the selected token.

4. **Enter Certificate Data.**

   Paste the certificate text in the text area provided. When you copy and paste the text, be sure to include the headers "Begin Certificate" and "End Certificate" — including the beginning and ending hyphens. You can also click Browse and select the `.DER` file manually.

5. **Provide Certificate Details.**

   Provide a nickname to be used for the certificate. Select the HTTP Listener from the available list for handling the secure requests. You can also select the self-signed certificate option.

6. **View Results.**

   This page provides you with a summary of selected options. Click Finish to complete the installation process.

Note – Using CLI

To install a certificate through CLI, execute the following command.

```
wadm> install-cert --user=admin --port=8989  --password-file=admin.pwd
--config=config1 --token=internal --cert-type=server --nickname=cert1 cert.req
```

where `cert.req` contains the certificate data.

See CLI Reference, `install-cert(1)`.

Requesting and Installing External Certificates

You can request and install certificates from other certificate authorities. A list of CAs are available in the industry. This section describes how to request and install external server certificates.

Perform the steps 1– 5, as described in the "To Request a Certificate" on page 93 section. Perform the following steps to complete the request for external certificate.

1. In the Certificate Type wizard, select the CA Signed Certificate option and click **Next**.
2. Review page is displayed. Verify the settings and click **Finish**.
3. Copy the Certificate Signing Requests (CSR) including the headers and click the **Close** button.
4. Go to the certificate authorities web site, complete the formalities to get the certificate signed by the authority.
5. Save the certificate in the local folder or copy the certificate from the web site.

To install the obtained certificate, perform the steps 1–3, as described in the "To Install a Certificate" on page 98. Perform the following steps to complete the installation for external certificates.

1. In the Enter Certificate Data page, paste the certificate or provide the path of the file that you have saved in the machine. Click the **Next** button.
2. Enter the nick name for the certificate and select the listener from the drop-down list. Click the Next button.
3. Review page is displayed. Click the **Finish** button to complete the installation.

For more information on setting a token pin, see "To Set the Token Password" on page 102.

Renewing a Certificate

You can renew an existing certificate by following these steps:

▼ To Renew a Certificate

1 Click Server Certificates tab > Certificate Name > Renew button.

2 Provide Token Information.

Enter the password for the token if required. Otherwise click Next to continue.

3 Update Certificate Details.

Review the certificate details and provide the validity period in months.

4 **Update Key Information.**

For key type, you can choose RSA or ECC. If the key type is RSA, the key size can be 1024, 2048 or 4098. If your key type is ECC you will also need to select a curve. Keep in mind that generating a new key pair takes time.

5 **View Summary.**

This page provides you with the summary of selected options. Click Finish to complete the renewal process.

Note – You must restart the administration server and node after the administration server certificates are renewed.

Deleting a Certificate

To delete a certificate, perform the following tasks:

▼ To Delete a Certificate

1 **Click Server Certificates tab.**

2 **Select the certificate.**

Select the certificate name from the certificate list.

3 **Delete certificate.**

Click the Delete button to delete the selected certificate.

Note – Using CLI

To delete a certificate through the CLI, execute the following command:

```
wadm> delete-cert --user=admin --port=8989  --password-file=admin.pwd
--token=internal --config=config1 cert1
```

See CLI Reference, `delete-cert(1)`.

Renewing Administration Server Certificates

To renew Administration Server certificates, with the nicknames `Admin-CA-Cert`, `Admin-Server-Cert`, and `Admin-Client-Cert` execute the `renew-admin-certs` CLI command. . This command also updates the nodes that are currently running and are accessible with the renewed certificates.

After executing this command, it is recommended that you restart the administration servers and nodes so that the new certificates can take effect. You must re-register a node if the node was offline (not running or was not accessible due to network problems) during the renewal of the certificates. To renew the administration server certificates, execute the following command.

```
wadm> renew-admin-certs --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --validity=120
```

See CLI Reference, `renew-admin-certs(1)`.

Managing Certificate Revocation Lists (CRL)

Certificate revocation lists (CRLs) makes known any certificate and key 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. CRLs are produced and periodically updated by a CA.

▼ To Install a CRL

To install a CRL obtained from a CA, perform the following steps:

1. **Obtain the CRL as a file from your CA.**
2. **Go to the configuration page in the administration console.**
3. **Click the Certificates > Certificate Authorities tab.**
4. **Click the Install CRL button.**
5. **Enter the full path name to the associated file.**
6. **Click OK.**

Note – If the CRL already exists in the database, a Replace Certificate Revocation List page will appear.

---

7. **You may need to click Deploy for changes to take effect.**

Note – Using CLI

To install a CRL through CLI, execute the following command.

```
wadm> install-crl --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 data/install-crl/ServerSign.crl
```

See CLI Reference, `install-crl(1)`.

▼ To Delete a CRL

1. **Go to the configuration page in the administration console.**
2. **Click the Certificates > Certificate Authorities tab.**
3. **Select the CRL entry and click Delete.**
4. **You may need to click Deploy for changes to take effect.**

Note – Using CLI

To delete a CRL through CLI, execute the following command.

```
wadm> delete-crl --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 issuer
```

See CLI Reference, `delete-crl(1)`.

Setting Password for the Internal Token

To set password for the internal PKCS11 token, perform the following tasks:

▼ To Set the Token Password

1. **Go to the configuration page in administration console.**
2. **Click the Certificates > PKCS11 Tokens tab.**
3. **Click the PKCS11 token name (default is internal).**
4. **Select the Token State checkbox.**

5. **Enter password information.**
6. **If you do not want to be prompted for password at instance startup, select the checkbox. Do not prompt for the new password at instance startup. Click OK.**
7. **The password will be saved in the configuration. To remove the password, perform the above steps and select Unset Password option.**

Note – Using CLI

To set the password for the internal PKCS11 token through CLI, execute the following command.

```
wadm> set-token-pin --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --token=internal
```

See CLI Reference, `set-token-pin(1)`.

Configuring SSL for the Server

SSL is the most popular standard for securing Internet communications and transactions. Web applications use HTTPS (HTTP over SSL), which uses digital certificates to ensure secure, confidential communications between server and clients. In an SSL connection, both the client and the server encrypt data before sending it, then decrypt it upon receipt.

When a Web browser (client) wants to connect to a secure site, an SSL handshake happens:

- The browser sends a message over the network requesting a secure session (typically, by requesting a URL that begins with `https` instead of `http`).
- The server responds by sending its certificate (including its public key).
- The browser verifies that the server's certificate is valid and is signed by a CA whose certificate is in the browser's database (and who is trusted). It also verifies that the CA certificate has not expired.
- If the certificate is valid, the browser generates a one time, unique session key and encrypts it with the server's public key. The browser then sends the encrypted session key to the server so that they both have a copy.
- The server decrypts the message using its private key and recovers the session key.

After the handshake, the client has verified the identity of the Web site, and only the client and the Web server have a copy of the session key. From this point forward, the client and the server use the session key to encrypt all their communications with each other. Thus, their communications are ensured to be secure.

The newest version of the SSL standard is called TLS (Transport Layer Security).

Use the command `create-cert-request` to generate a request and send the request to your CA. Later, when you receive the certificate from the CA you'll need to install it using the `install-cert` command. If you have a key and a certificate in a Java keystore which you're looking to migrate, use the command `migrate-jks-keycert`. For a development/test server, the easiest way to get going is to generate a self-signed certificate using the command `create-selfsigned-cert`.

```
wadm> create-selfsigned-cert --server-name=hostname --nickname=MyServerCert
--token=internal
```

Check the man pages for more options and examples.

With the certificate installed, you need a listener on a port which will have SSL enabled.

```
wadm> create-http-listener --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --listener-port=18003 --config=config1 --server-name=config1.com
--default-virtual-server-name=config1_vs_1 config1_ls_1
```

Next enable SSL for the listener and associate the listener with the nickname of the certificate.

```
wadm> set-ssl-prop --http-listener=http-listener-ssl enabled=true
wadm> set-ssl-prop --http-listener=http-listener-ssl server-cert-nickname=MyServerCert
```

After this setup, deploy the configuration and start the instance.

```
wadm> deploy-config config_name
wadm> start-instance --config config_name hostname
```

Note – Using Administration Console

To create a self-signed certificate through Administration Console, perform the tasks as mentioned in "Requesting a Certificate" on page 93 and select 'Self-Signed Certificate' as the certificate type.

Enabling SSL Ciphers for a Configuration

To enable SSL Ciphers for a configuration, execute the following command.

```
wadm> enable-ciphers --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --http-listener=http-listener-1
--cipher-type=ssl2 SSL_RC4_128_WITH_MD5
```

See CLI Reference, `enable-ciphers(1)`.

Enabling Security For HTTP Listener

Note – Security can be enabled for the HTTP listener only when there are available installed certificates.

Once you have a certificate, you can associate the certificate with a HTTP Listener and thus secure the server.

Encryption is the process of transforming information so it is meaningless to anyone except the intended recipient. Decryption is the process of transforming encrypted information so that it is meaningful again. Web Server includes support for SSL and TLS 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.

Web Server 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, to transmit certificates, and to 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.

Click the **Configurations > HTTP Listeners > Security tab** to edit the HTTP Listeners security settings. The following table lists the properties that you can configure in this page.

TABLE 6–1 HTTP Listener Security Properties

Property	Description
Name	Name of the HTTP Listener.
Security	Enable/Disable security for the selected HTTP Listener.
Certificate	Select the server certificate from the available certificates. You should have installed either a RSA or ECC certificate for performing this action.
Client Authentication	Specifies whether the client authentication is required or optional. Select False option to disable client authentication.
Authentication Timeout	Timeout after which client authentication handshake fails. [0.001–3600]. The default value is 60 seconds.
Maximum Authentication Data	Maximum amount of authentication data to buffer. [0–2147.0483647.0]. The default value is 104857.06.
SSL Version 2/Version 3	Enable/Disable SSL Version 2/ SSL Version 3.
TLS	Enable/Disable TLS. Detect version rollbacks is enabled by default. This configures the server to detect man-in-the-middle version rollback attack attempts. Disabling this may be required for interoperability with some clients that incorrectly implement the TLS specification.
SSL3/SSL2/TLS 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 HTTP Listener for the Administration Server. The default settings allow the most commonly used ciphers. Unless you have a compelling reason why you do not want to use a specific cipher suite, you should allow them all.

◆ ◆ ◆ CHAPTER 7

Controlling Access to Your Server

You can protect resources that reside on your web server through several security services and mechanisms, including authentication, authorization, and access control. This chapter describes some of the supported mechanisms for controlling access to your Web Server .

What is 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 (Web application security, htaccess, Authentication Realm and more) and services.

Access control enables you to determine:

- Who can access your Administration Server
- Which applications 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. 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, the server checks if any ACLs are configured for

that virtual server. If ACLs are found that apply for the current request, the server evaluates their ACEs to determine whether access should be granted or denied.

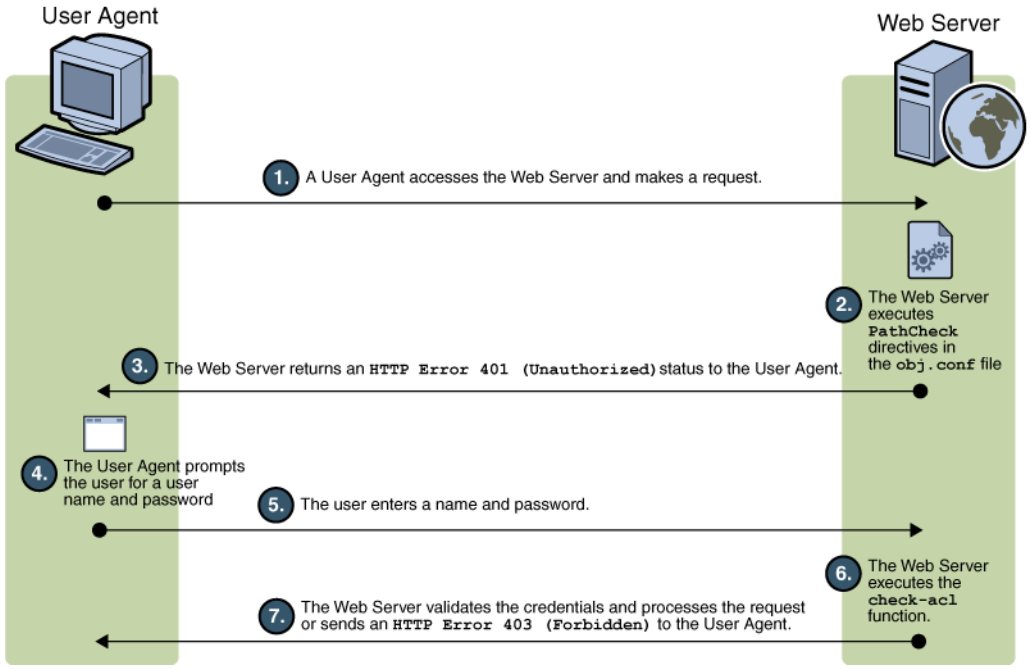
You allow or deny access based on:

- 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)

How Access Control Works

When the serverreceives a request for a page, the server uses the rules in the ACL file to determine whether 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.

Note – If there is more than one ACL that matches, 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.



The preceding figure depicts how access control works in Web Server . The user agent (client) accesses the Web Server, and then the Web Server executes `PathCheck` directives in `obj.conf` file. The Web Server returns an HTTP 401 (unauthorized) to the client. The client prompts the user for authentication. In case if the client is a browser, a login dialog box appears. The user enters the login information. The Web Server executes an internal `check-acl` function. The Web Server validates the user credentials and processes the request.

Setting Up Access Control for User-Group

You can limit access to 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 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 changing the method to SSL.

User-Group authentication is performed by Web Server by reading entries in the user group database. 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 are:

- Default
- Basic
- Digest

User-Group authentication requires users to authenticate themselves before gaining access to the server, or the files and directories on your web site. The authentication process involves users verifying their identity by entering a username and password, using a client certificate. Client certificates are required only for SSL communication.

Default Authentication

Default authentication is the preferred method of authentication. The Default setting uses the default method in the `server.xml` file, or "Basic" if there is no setting in `server.xml`. If you check Default, the ACL rule doesn't specify a method in the ACL file. Choosing Default enables you to easily change the methods for all ACLs by editing one line in the `obj.conf` file.

Basic Authentication

Basic authentication requires users to enter a username and password to access your web server or web site. Basic authentication is the default setting and in order to use it, you must create and store a list of users and groups in an LDAP database, such as the Sun Java System Directory Server, or in a file. You must use a directory server installed on a different server root than your 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 un-encrypted text across the network and means that the network packets could be intercepted, and the username and password can be pirated. Basic authentication is most effective when combined with SSL encryption, Host-IP authentication, or both. Using Digest Authentication avoids this problem.

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.
- Maps the certificate to a user's entry using the `certmap.conf` file, if the certificate is from a trusted certificate authority (CA).
- 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 to control 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.

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.

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 require modification to the `certmap.conf` file. If you choose to use client certificates, you should increase the value of the `AcceptTimeout` directive in the `magnus.conf` file.

Digest Authentication

The server can be configured to perform digest authentication using either an LDAP-based or a file-based directory service.

Digest authentication enables 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 Java System Directory Server includes a reversible password plug-in using a symmetric encryption algorithm to store data in an encrypted form, that can later be decrypted to its original form. Only the Directory Server holds the key to the data.

For LDAP-based digest authentication, you need to enable the reversible password plug-in and the digestauth-specific plug-in included with the server. To configure your web server to process digest authentication, set the `digestauth` property of the database definition in `dbswitch.conf`.

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 7–1 Digest Authentication Challenge Generation

ACL Method	Digest Authentication Supported by Authentication Database	Digest Authentication Not Supported by Authentication Database
"default" none specified	digest and basic	basic
"basic"	basic	basic

TABLE 7–1 Digest Authentication Challenge Generation *(Continued)*

ACL Method	Digest Authentication Supported by Authentication Database	Digest Authentication Not Supported by Authentication Database
"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 the realm 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.

Setting Access Control for the 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 host names 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.

Enabling DNS degrades the performance of the 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, append `iponly=1` to `AddLog fn="flex-log" name="access"` in your `obj.conf` file:

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

Configuring the ACL User Cache

By default, the 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 the server every time you make changes to the LDAP entries. For example, if this value is set to 120 seconds, the 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 *NSAPI Developer's Guide*.

Setting ACL Cache Properties

To set ACL cache properties through CLI, execute the following command.

```
wadm> set-acl-cache-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 property=value
```

See CLI Reference, `set-acl-cache-prop(1)`.

The valid properties you can set are:

- `enabled` — Indicates whether the server caches the file content and meta information. The default value is true.
- `max-age` — The maximum amount of time (in seconds) to cache the file content and meta information. The range of values is 0.001 to 3600.
- `max-groups-per-user` — The maximum number of groups per user for which the server will cache the membership information. The range of values is 1 to 1024.
- `max-age` — The maximum amount of time (in seconds) to cache the authentication information. The range of values is 0.001 to 3600.

Configuring Access Control

The server 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`.

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.

The core ACLs supported by the server are three types of authentication: basic, SSL, and digest.

To edit access control settings, perform the following tasks:

1. Click the **Configurations** tab and select the configuration.
2. Click the **Security sub tab > Access Control sub tab**.
3. Click the **Add ACL** button to add a new ACL or click existing ACL to edit the settings.

Adding an Access Control List (ACL)

The following section describes the process of adding a new ACL to the configuration.

1. Click the **Configurations** tab and select the configuration.
2. Click the **Access Control sub tab > Access Control Lists sub tab**.
3. Click the **New** button to add a new ACL.

Configure the following parameters:

TABLE 7–2 ACL Parameters

Parameter	Description
Resource	Named/URI/Path. Select the type of resource you need to set access restriction and specify the value. Example for URI resource — "/sales". Example for Path resource — "/usr/sun/server4/docs/cgi-bin/*".
Authentication DB	*Authentication Database* lets you select a database the server will use to authenticate users. The default is **keyfile**
Authentication Method	1. **Basic** — uses the HTTP Basic method to obtain authentication information from the client. The username and password are only encrypted over the network if SSL is turned on for the server. 2. **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. 3. **Digest** — uses an authentication mechanism that provides a way for a browser to authenticate based on username and password without sending the username and password as clear text. The browser uses the *MD5* algorithm to create a digest value using the user's password and some information provided by the Web Server. Note that in order to use Digest the underlying auth-db must support digest as well. This means either a File auth-db using digestfile or an LDAP auth-db must be present if the Digest Authentication Plug-in has been installed 4. **Other** — uses a custom method created using the access control API.
Prompt for Authentication	**Prompt for Authentication** option enables 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 browser, the user will see the first 40 characters of the prompt. Web browsers typically cache the username and password, and associate them with the prompt text. When the user accesses server files and directories with 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.
Denied Access Response	Specify the response action when an access to a resource is denied. 1. Respond with default message — Select this option to display the standard access denied message from the server. 2. Respond with URL — Select this option to forward the request to any other external URL or error page.

Note – Using CLI

To add an ACL through the CLI, execute the following command.

```
wadm> set-acl --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --vs=config1_vs_1 --config=config1
--aclfile=aclfile1
```

See CLI Reference, `set-acl(1)`.

Adding an Access Control Entry (ACE)

The section describes the process of adding a new Access Control Entry (ACE) for the selected configuration.

1. Click the **Configurations** tab and select the configuration.
2. Click the **Access Control sub tab > Access Control List sub tab**.
3. Click the **New** button.
4. Click the **New** button under Access Control Entry.

Configure the following ACE parameters:

TABLE 7–3 ACE parameters

Parameter	Description
Access	■ **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.
Users	1. **Anyone** — No authentication. Grants access to everyone. 2. **All in the Auth DB** — Grants access to all users specified in the authentication database. 3. **Only the following in the Auth DB** — Restrict access to selected users from the authentication DB. You can query the authentication DB based on common attributes like First name, Last name and Email address.
Groups	With group authentication, users are prompted to enter a username and password before they can access the resource specified in the access control rule. Use this option to restrict access to specific groups.

TABLE 7–3 ACE parameters *(Continued)*

Parameter	Description
From Host	You can restrict access to the Administration Server or your web site based on which computer the request comes from. ■ Anyplace enables access to all users and systems. ■ Only from enables 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 will 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 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`.

TABLE 7–3 ACE parameters *(Continued)*

Parameter	Description
Rights	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 enables 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** enables users to view files, including includes the HTTP methods GET, HEAD, POST, and INDEX. ■ **Write** enables 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** enables users to execute server-side applications, such as CGI programs, Java applets, and agents. POST maps to execute right only. ■ **Delete** enables users who also have write privileges to delete files or directories. ■ **List** enables users to access lists of the files in directories that do not contain an `index.html` file. ■ **Info** enables users to receive information about the URI, for example `http_head`.
Continue	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 will 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.

Using .htaccess File

The server supports `.htaccess` dynamic configuration files. You can enable `.htaccess` files either through the user interface or by manually changing the configuration 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". Use SSL client authentication via the standard server access control, and also require HTTP "Basic' authentication via an `.htaccess` file.

If you enable .htaccess files, the server checks for .htaccess files before serving resources. The server looks for .htaccess files in the same directory as the resource and in that directory's parent directories, up to and including the document root. For example, if the Primary Document Directory is set to /sun/server/docs and a client requests /sun/server/docs/reports/index.html, the server will check for .htaccess files at /sun/server/docs/reports/.htaccess and /sun/server/docs/.htaccess.

Note that the Server's Additional Document Directories and CGI Directory functionality enables an administrator to define alternate document roots. The existence of alternate document roots affects .htaccess file processing. For example, consider a server with the Primary Document Directory set to /sun/server/docs and a CGI program at /sun/server/docs/cgi-bin/program.cgi. If you enable CGI as a File Type, the server will evaluate the contents of both /sun/server/docs/.htaccess and /sun/server/docs/cgi-bin/.htaccess when a client issues a request for the CGI program. However, if you instead configure a CGI Directory at /sun/server/docs/cgi-bin, the server will inspect /sun/server/docs/cgi-bin/.htaccess but not /sun/server/docs/.htaccess. This occurs because specifying /sun/server/docs/cgi-bin as a CGI Directory marks it as an alternate document root.

Preventing Denial-of-Service Attack

Denial-of-Service (DoS) attack is an explicit attempt to prevent legitimate users from using a service by some malicious users of the Server. Such an attack can be launched by sending continuous requests to the server for a particular web resource.

Web Server can detect DoS attack by monitoring frequently accessed URI and denying requests if the request frequency is high.

The following sections describes how you can prevent DoS attacks at the virtual server level.

Limiting Requests to the Server

You can now tweak the server to prevent Denial-Of-Service attacks by configuring request limits and monitoring maximum number of connections per virtual server. Configuring some of these values may affect the server's performance.

To configure request limits for the server, click **Configuration > Virtual Servers > Server Settings > Request Limits**. Configure the parameters listed in the following table.

TABLE 7–4 Configuring Request Limit

Parameter	Description
Request Limits	Enable/Disable request limits for this virtual server. Request limits option is disabled by default.
Maximum Connections	Maximum number of concurrent connections allowed for this virtual server.
Maximum RPS	Maximum number of requests allowed from a client per second.
RPS Compute Interval	The time interval in which the average request per second (RPS) is calculated. Default values is 30 seconds.
Continue Condition	Determines what condition must be met in order for a blocked request type to become available again for servicing. **silence** — Refused requests must fall to zero (over a subsequent interval) for service to resume. **threshold** — Refused request rate must fall below RPS threshold for service to resume. The default values is threshold.
Error Code	The HTTP status code to use for blocked requests. The default code is HTTP 503 — Service Unavailable.
Monitor Attribute	An optional request attribute to monitor.

Note – Using CLI

To limit the requests to the server through CLI, execute the following command.

```
wadm> enable-request-limits --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
```

See CLI Reference, `enable-request-limits(1)`.

▼ To Limit the Maximum Number of Connections

You can limit the maximum number of concurrent connections. If a matching request is received while there are at least the specified number of requests being processed, the request is rejected. Note that the rejection of request only occurs for that particular time. As soon as concurrent requests drops below this limit new requests will be processed.

1 **Click the Configuration tab.**

2. **Select your configuration from the list.**
3. **Select your virtual server under the Virtual Server tab.**
4. **Click Server Settings > Request Limits.**
5. **Enter a value for Maximum Connections section.**

Preventing Cross Site Scripting Attacks

Cross site scripting (XSS) is a common security problem of web applications where an attacker gains access to the users current web browser session.

Web sites today are highly complex containing huge amount of dynamic content, which are generated through web applications, delivering different output depending on the requirement of the user. An attacker may inject malicious data or scripting code into pages generated by the web application and it may appear as a valid content from a trusted site. Such HTML pages pose security risk, if inputs are not validated by the web application. In the user's generated output browser page, the scripting code is executed and facilitates the transfer of sensitive data to the attacker. Through an XSS attack, confidential information like ID, password, security access information and credit card information, can be obtained.

Cross site scripting thus pose an immense risk to individuals or an entire organization. Input validation at all application points that accept data on the server side is one way of solving this problem.

In Web Server 7.0 XSS prevention is accomplished through the addition of `sed-request` filter and entity encoding, using entities like `<` and `>` which encodes < and > characters.

An input stage filter, the `sed-request` applies `sed` edit commands to an incoming request.

```
Input fn="insert-filter" filter="sed-request" sed="script"
```

where `script` is the actual `sed` script you want to run on request body.

To configure XSS prevention, add the below information in the `obj.conf` file's default object:

```
Input fn="insert-filter"
method="POST"
filter="sed-request"
sed="s/(<|%3c)/\\</gi"
sed="s/(>|%3e)/\\>/gi"
```

For information about `sed-request`, see "sed-request" in *Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference*.

◆ ◆ ◆ CHAPTER 8

Managing Users and Groups

This chapter describes how to add, delete, and edit the users and groups who can access your server.

Accessing Information About Users and Groups

The Administration Server provides access to your application data about user accounts, group lists, access privileges (ACL), 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 the Sun Java System 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.

About Directory Services

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

In Web Server 7.0, 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 createa directory service, the `server.xml` file is updated with the directory service details.

Types of Directory Services

The different types of directory services supported by Web Server 7.0 are:

- **LDAP** — Stores user and group information in an LDAP-based directory server.
- **Key File** — 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 key file are used for authorization and authentication by the `file` realm alone; these bear no relationship to system users and groups.

  The key file format can only be used when the intent is to use HTTP Basic authentication.
- **Digest File** — Stores user and group information based on encrypted username and password.

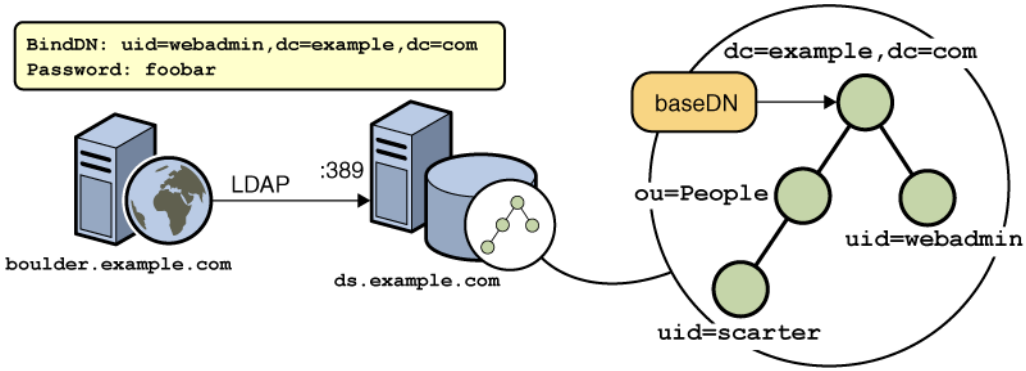
  The digest file format is meant to support using HTTP Digest authentication. It does, however, also support Basic authentication, so it can be used for both authentication methods.

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

Understanding Distinguished Names (DNs)

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.

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 preceding figure shows a sample DN representation. 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.

Working With the Authentication Database

The *Authentication Database*, also referred to as auth-db, represents a database of known users and the mechanism for authenticating client requests against that database. The server can have multiple auth-db entries configured at the same time and these may be of the same type. The auth-db user databases are used by the ACL processing module.

The server supports the following authentication databases:

1. **LDAP** — The user data is stored in an LDAP directory server, such as the Sun Java System Directory Server.
2. **File** — The user data is stored in a disk file. This auth-db is particularly convenient for development or small deployments where no centralized user management is available (or desired). The file auth-db supports several different file formats:
   a. keyfile — The keyfile format stores a list of users (and optional group memberships for each user). The password is stored as a one-way (unrecoverable) hash. This is the default format.
   b. digestfile — The digestfile is very similar to the keyfile and also supports the HTTP Digest authentication method.
   c. htaccess — The htaccess is a legacy format and should never be used for new installations or adding new users.
3. **PAM** — PAM is the new auth-db supported by Web Server . The PAM auth-db delegates the authentication to the Solaris PAM stack, this enables existing Solaris users on the web server system to authenticate to the web server as well.

Note – PAM auth-db is only supported in Solaris 9 and 10 (or higher) and the web server instance must be running as root.

Creating an Authentication Database

FTo create an authentication database through the Administration Console, click **Configurations > Configuration Name > Access Control > Authentication Databases > New** button. Check out the Administration Console Inline help for field descriptions. Based on the selected Authentication Database, the fields change. For example, for PAM based Authentication DB, only the authentication DB name is required.

The options required to create an Authentication Database are enumerated here:

LDAP	■ Authentication Database Name ■ Host Name ■ Port ■ Base DN
Key FIle	■ Authentication Database Name ■ File Path
Digest File	■ Authentication Database Name ■ File Path
PAM	■ Authentication Database Name

To create an authentication database through CLI, execute the following command.

```
wadm> create-authdb --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--url=ldap://ldapserver.com:20002/dc=xxx,dc=sun,dc=com LDAP1
```

See CLI Reference, `create-authdb(1)`.

In the previous example, a URL has been specified for the authentication database. The type of authentication database is specified in this URL scheme. For example, `ldap://ds.example.come/dc=example,dc=com` configures an LDAP directory server as an authentication database.

Setting Up Users and Groups

The Administration Server enables editing user accounts, group lists, access privileges, organization units, and other user- and group-specific information for both LDAP and File auth-db types.

▼ To Add a User

1. **Click the Configuration tab to see a list of users and select the configuration you need.**
2. **Click the Access Control > Users tab.**
3. **Click the New button.**
4. **Add User Information.**

   Enter the user id and password. Optionally enter the group which the user belongs to. The user ID must be unique. In case of LDAP based authentication DB, 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.

Note – Using CLI

To create a user through CLI, execute the following command.

```
wadm> create-user --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --authdb=KEYFILE1 --full-name=keyfile-config1-u1
keyfile-config1-u1
```

See CLI Reference, `create-user(1)`.

▼ To Add a Group

1. **Select the configuration from the configuration list. Click Configurations tab to get the list.**
2. **Click Access Control > Groups tab.**
3. **Click the New button.**
4. **Enter the Group Name.**
5. **From the Add Users To Group section, search and add existing users to the group.**

Note – Creating a group in an authentication database like keyfile or digestfile requires atleast one user to be specified.

Note – Using CLI

To create a group through CLI, execute the following command.

```
wadm> create-group --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --authdb=LDAP1 group1
```

See CLI Reference, `create-group(1)`.

To Delete a User

1. **Select the configuration from the configuration list. Click Configurations tab to get the list.**
2. **Click the Access Control > Users tab.**
3. **Select the authentication database from which you need to delete the user.**
4. **Enter the User ID in the Search Users text box and click the Search button.**
5. **Select the user from the UserID column and click the Delete button.**

Caution – Deleting users from a keyfile or digestfile authentication database will delete any associated groups even if the groups contain no members. Groups without members are not allowed in keyfile or digestfile authentication databases.

Note – Using CLI

To delete a user through CLI, execute the following command.

```
wadm> delete-user --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config config1 --authdb KEYFILE1 user1
```

See CLI Reference, `delete-user(1)`.

To Delete a Group

1. **Select the configuration from the configuration list. Click Configurations tab to get the list.**
2. **Click the Access Control > Groups tab.**
3. **Select the authentication database from which you need to delete the group.**
4. **Enter the Group Name in the Search Users text box and click the Search button.**
5. **Select the group from the Group Name column and click the Delete button.**

Note – Deleting a group does not delete the users belonging to the group. You have to delete the users manually or reassign groups.

Note – Using CLI

To delete a group through CLI, execute the following command.

```
wadm> delete-group --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config config1 --authdb LDAP1 group1
```

See CLI Reference, `delete-group(1)`.

Static and Dynamic Groups

A group is a set of objects in an LDAP database. In Web Server 7.0 a group consists of users who share a common attribute. For example, the set of objects might be the number of employees who work in the marketing division of your company. These employees might belong to a group called Marketing.

For LDAP services, there are two ways to define 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`. In 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. In 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=Airius.com??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 dynamic groups, members can share a common attribute from a certificate if you use the `memberCertDescription`. Note that these attributes will only work if the ACL uses the SSL method.

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

Static Groups

For LDAP services, the Administration Server enables you to create a static group by specifying the same group attribute in the DNs of any number of users. A static group 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 topmost entry.

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.

For LDAP services, 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.

How Web Server Implements Dynamic Groups

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 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 – 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, Web Server checks all dynamic groups in the database's baseDN. 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 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 8–1 Dynamic Groups: Required Parameters

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

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

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

◆ ◆ ◆ CHAPTER 9

Managing Server Content

This chapter describes how you can configure and manage content across virtual servers.

Configuring Document Directories

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.

You can create a document directory which is in addition to the primary document directory. By doing this, you can let someone manage a group of documents without giving them access to your primary document root.

▼ To Create a Document Directory

1. **Click the Configuration tab and select the configuration needed.**
2. **Click the Virtual Servers tab to get the list of configured virtual servers for the selected configuration and select the virtual server for which you need to add a new document directory.**
3. **Click Content Handling > Document Directories tab.**
4. **Click New and configure the following parameters:**
   - **URL Prefix** — URI prefix that has to be mapped to a directory.
   - **Directory Path** — Absolute server path and a valid directory for storing documents.

Note – When deploying a configuration across instances in a cluster, the document directories and documents are not deployed across the nodes in a cluster. The document directories are either NFS mounted onto the nodes or are available on the nodes. Only `config`, `deployed webapps`, `using add-webapp`, and `generated` directories containing user initiated pre-compiled JSP files created during `add-webapp`, are propagated.

Note – Using CLI

To create a document directory through CLI, execute the following command.

```
wadm> create-document-dir --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
--uri-prefix=/config1_uri --directory=../docs1
```

See CLI Reference, `create-document-dir(1)`.

Changing the 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/html
■ text/richtext	■ image/tiff
■ image/jpeg	■ image/gif
■ application/x-tar	■ application/postscript
■ application/x-gzip	■ audio/basic

▼ To Change the Default MIME Type

1. **Click the Configurations tab and select the configuration needed.**
2. **Click the Virtual Servers tab to get the list of configured virtual servers for the selected configuration.**
3. **Click Content Handling > General tab.**
4. **Change Default MIME Type value under Miscellaneous section**

Note – Using CLI

To create a MIME type through CLI, execute the following command.

```
wadm> create-mime-type --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --extensions=sxc application/sxc
```

See CLI Reference, `create-mime-type(1)`.

You need not create a separate MIME types file for each virtual server. Instead, you can create as many MIME types files as you require and associate them with a virtual server. By default, one MIME types file (mime.types) exists on the server and cannot be deleted.

Enabling Directory Listing

When a web browser is pointed to a directory on your web site that does not have an index.html file or a welcome file, the files in that directory cannot be listed on a web page. You can turn on directory listing for a virtual server by executing the following command:

```
enable-directory-listing --user=admin --host=serverhost
--password-file=../admin.passwd  --port=8989 --ssl=true --no-prompt
--rcfile=null --index-style=simple --config=config1 --vs=vs
```

The –`index-style` field denotes the type of directory index to generate if the server cannot find one of the index file names specified. If your server is outside of firewall, turn directory listing on. This format includes a graphic that represents the type of the file, the last modified data and the file size.

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.

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:

▼ Configuring Document Directories

1. **From the virtual server page, click the Content Handling tab.**
2. **Click Document Directories.**
3. **Under User Document Directories, 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.**

7 **Click Save.**

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 which user accounts are able to publish content by means of user document directories. To restrict a user's publishing rights, 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, 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 recorded 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.

Setting Up URL Redirection

URL redirection enables you to redirect document requests from one HTTP URL to another HTTP URL. Forwarding URLs, also known as redirection, is a method in which the server tells a user that a URL has changed. URLs may change when files are moved to another directory or server. You can use redirection to seamlessly send request for a document on one server to a document on another server.

For example, if you forward `http://www.sun.com/info/movies` to the prefix `film.sun.com`, the URL `http://www.sun.com/info/movies` redirects to `http://film.sun.com/info/movies`.

Sometimes you may want to redirect requests for all the documents in one sub-directory to a specific URL. For example, if you had to remove a directory because it was causing too much traffic, or because the documents were no longer to be served for any reason, you can 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` can be redirected to `http://www.sun.com/explain.html`.

You can set URL redirection at the virtual server level.

To configure URL redirection, perform the following steps:

1. Click the **Configurations tab** and select the configuration from the configuration list.
2. Click the **Virtual Servers sub tab** and select the virtual server from the virtual server list.
3. Click the **Content Handling sub tab** and then the **URL Redirects sub tab.**
4. Click **New** to add a new URL redirect rule.
5. Provide necessary values for the fields. Click **OK** . Click **Deploy** for the configuration if needed.

The following table describes the parameters required while adding a new URL Redirect rule.

TABLE 9–1 URL redirect Parameters

Parameter	Description
Source	**URI Prefix** — URI from which the requests should be redirected. All HTTP requests to this URI pattern will be redirected to the URL specified by the Target URL. **Condition** — Instead of providing a URI prefix as the source, you can also use regular expression in the condition text field. For example, if you need to redirect requests to a particular URL, if the browser is Mozilla, then type `$browser = "Mozilla"` in the condition field. Another valid example is `$browser =~ "MSIE"`. The Web Server includes a set of variables predefined by the server, as well as the capability for you to define custom variables. In our example, `browser` is a pre-defined variable. For a list of available pre-defined variables, see "Predefined Variables" in *Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference*. You can define custom variables in the server.xml file using the variables element and then refer to those variables from this condition text. Administration Console supports only `<If>` tag with redirect SAF for `NameTrans` directive with all the variables, expression literals, expression functions and expression operators. For more information on variables, operators and expression, see the *Administrator's Configuration File Reference Guide*.
Target URL	URL to which the requests should be redirected. All HTTP requests from the URL specified in the From URL will be redirected to this URL.
URL Type	Fixed. **Enabled/Disabled**. Fixed URLs are static URLs like a link to an HTML page. Non-Fixed URLs are dynamic URLs with request parameters or URLs with just prefixes.

Note – Using CLI

To add a new URL redirection rule through CLI, execute the following command.

```
wadm> create-url-redirect --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --no-ssl --config=config1 --vs=config1_vs_1 --uri-prefix=/redirect
--target-url=http://www.cnet.com
```

See CLI Reference, `create-url-redirect(1)`.

URL Redirection Using Regular Expression

Web Server is enhanced to support regular expressions (also known as Patterns) and request time parameter interpolation in configuration files. In addition, wildcard pattern matching support is extended to server.xml. URL redirecting is implemented as an SAF. The redirect SAF lets you redirect URIs that match a certain prefix. You can specify the prefix using the `from` parameter. You can specify the URL to redirect using the `url` or `url-prefix` parameters. In Web Server Web Server, the `from` parameter is optional. If `from` is omitted, all URIs are redirected.

In the obj.conf file, SAF parameters are supported with new `<If>`, `<Elseif>` and `<Else>` tags. These tags contain directives. Using these tags, you can define conditions under which the directives are executed. These tags can also be used to dynamically generate SAF parameters.

Web Server offers URL rewrite capability that is a super set of Apache HTTP server's `mod_rewrite` module. Unlike Apache's `mod_rewrite` function, `<If>` tag provides the following functionality:

- It can manipulate URI, path, header fields and response bodies.
- It works at any stage of request processing.
- It works with any SAF, including 3rd party plug-ins.

Consider the following directive:

```
NameTrans fn="redirect"
                   from="/site1"
                url="http://site1.mycompany.com"
```

The above directive can be rewritten using regular expression as follows:

```
<If $uri =~ '^/site1'>
           NameTrans fn="redirect"
           url="http://site1.mycompany.com"
</If>
```

In the above example, note the usage of regular expression instead of the `from` parameter. If you need to redirect all requests for `/site1/*` to `http://site1.mycompany.com/*/index.html` note this technique:

```
<If $uri =~ '^/site1/(.*)'>
           NameTrans fn="redirect"
           url="http://site1.mycompany.com/$1/index.html"
</If>
```

Here, the <If> tag assigns whatever value matches (.*) to the variable $1. The $1 in the `url` parameter is dynamically replaced with the value from the original request. That means the above obj.conf example will cause a request for `/site1/download` to be redirected to `http://site1.mycompany.com.com/download/index.html`.

The combination of <If> and redirect offers some of the flexibility of mod_rewrite. However, unlike mod_rewrite, <If> can be used for things other than redirecting and rewriting URLs. <If> can also be used in conjunction with any SAF, including third party plug-ins.

The previously mentioned method configures a 302 Moved Temporarily redirect. In Web Server , you can also add a status="301" parameter to indicate that you need a 301 Moved Permanently redirect instead

```
NameTrans fn="redirect" from="/path" url="http://server.example.com" status="301"
```

What is Not Supported

Support for <If>, <Else> and <ElseIf> tags is limited in the Administration Infrastructure (Administration Console and CLI). Though obj.conf file supports these tags with any directives, variables, SAFs, expression literals, expression functions and expression operators, the Administration Infrastructure supports only <If> tags with redirect SAF for NameTrans directive with all the variables, expression literals, expression functions and expression operators.

For example, you can configure:

```
<If $browser =~ "MSIE">
    NameTrans fn = "redirect" url="/msie.html"
</If>
```

But you cannot configure:

```
If $browser =~ "MSIE">
    NameTrans fn = "redirect" url="/msie.html"
</If>
<Else>
    NameTrans fn="redirect" url="/other.html"
</Else>
```

Note – You can use the get-config-file and set-config-file CLI commands to make use of more complicated expressions using <If>, <ElseIf> and <Else>.

See, get-config-file(1) and set-config-file(1).

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 CGI execution. These processes are created only during the first access to a CGI. Their number varies depending upon the CGI load on the server. Do not kill these CGIStub processes. They disappear when the server is stopped.

For more information see the discussion regarding `MinCGIStub`, `MaxCGIStub`, and `CGIStubIdleTimeout` in Web Server's *Performance Tuning and Sizing Guide*.

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.

Note – In order to run the command-line utilities, you need to manually set the `Path` variable to include *server_root*`/bin/https/bin`.

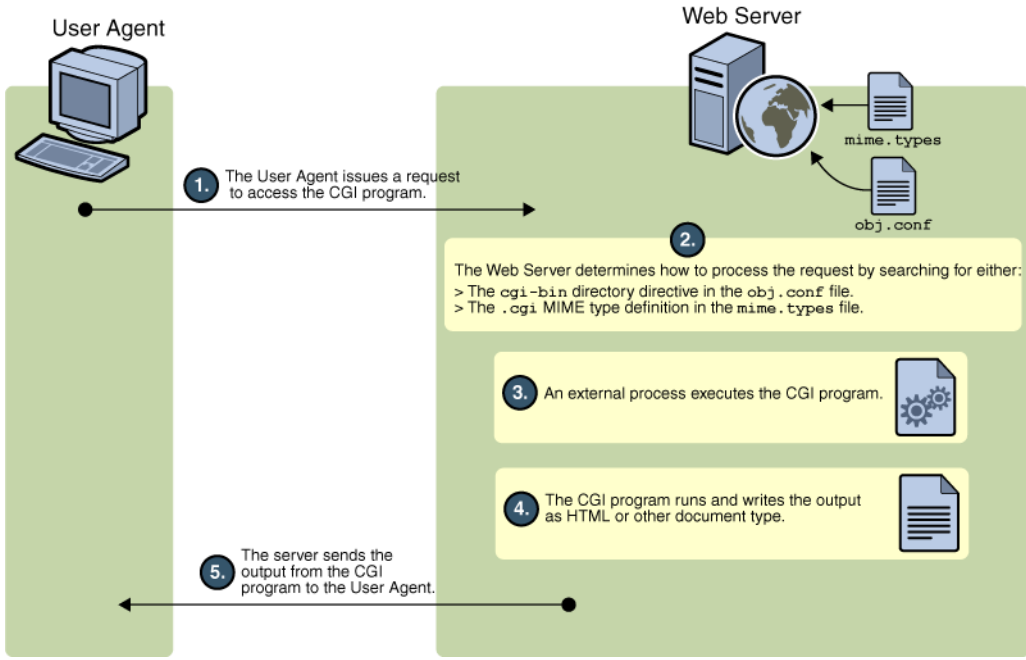
Regardless of the programming language, all CGI programs accept and return data in the same manner. For information about writing CGI programs, see the following sources of information:

- Web Server *Developer'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`

The following figure describes how a CGI request is processed in Web Server 7.0:



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.

Configuring CGI Subsystem for Your Server

Web Server enables you to add CGI document directories using the administration console GUI.

To add a new CGI document directory, perform the following tasks:

1. Click the **Configurations** tab and select the configuration from the configuration list.
2. Click the **Virtual Servers** sub tab and select the virtual server from the virtual server list.
3. Click the **Content Handling** sub tab and **CGI** sub tab.
4. Click **New** to add a new CGI document directory.
5. Provide necessary values for the fields. Click **OK** . Click **Deploy** for the configuration if needed.

The following table describes the fields required while adding a new CGI document directory.

TABLE 9–2 CGI Parameters

Parameter	Description
Prefix	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*
CGI Directory	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. **Note –** The URL prefix you specify can be different from the real CGI directory.
User	Specify the name of the user to execute CGI programs as.
Group	Specify the name of the group to execute the CGI programs as.

TABLE 9–2 CGI Parameters *(Continued)*

Parameter	Description
Chroot	Specify the directory to chroot to before execution begins.
Nice	Specify a nice value, 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.

To remove an existing CGI directory, select the CGI directory and click Delete. To change the URL prefix or CGI directory of an existing directory, click the directory link.

Copy your CGI programs into the directories you've specified. Remember that any files in those directories will be processed as CGI files, so do not put HTML files in your CGI directory.

To specify CGI as a file type, perform the following tasks:

1. Click the **Configurations** tab and select the configuration from the configuration list.
2. Click the **Virtual Servers** sub tab and select the virtual server from the virtual server list.
3. Click the **Content Handling** sub tab and **CGI sub tab.**
4. Select **CGI** as file type radio button.

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.

Note – Using CLI

You can create a CGI directory that contains the CGI programs that will be processed by your server. CGI programs are in a certain file type such as .cgi, .exe, or .bat. The programs can be located in any directory in or under the document root directory.

To add a CGI directory through CLI, execute the following command.

```
wadm> create-cgi-dir --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --uri-prefix=/config1_urlprefix
--directory=/cgi-dir
```

See CLI Reference, `create-cgi-dir(1)`.

Downloading Executable Files

If you're using `.exe` as a CGI file type, you cannot download `.exe` files as executable files.

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). 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.

Installing Shell CGI Programs for Windows

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 receives 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 Web Server.

Note – For information on setting Windows file extensions, see Windows documentation.

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 receives 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.

To add a custom error page, follow these steps:

1. Click the **Configurations** tab and select the configuration from the configuration list.
2. Click the **Virtual Servers** sub tab and select the virtual server from the virtual server list.
3. Click the **Content Handling** sub tab and then the **Error Pages sub tab.**
4. Click **New** to add a custom error page.

   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** to return to the error pages list.

Note – Using CLI

To customize error pages through CLI, execute the following command.

```
wadm> set-error-page --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --code=500
--error-page=/server-error-uri-new
```

See CLI Reference, `set-error-page(1)`.

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 for a document, a set of documents, or a directory by selecting a resource and entering a character set for that resource.

Most browsers can use the MIME type `charset` parameter in HTTP to change its character set. If the server includes this parameter in its response, browsers 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 some common browsers are specified in RFC 17.000. The names that begin with `x-` are not recognized by common browsers:

■ `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`	■ `ibm367.0`
■ `cp367.0`	

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

■ `latin1`	■ `iso_8859-1`
■ `iso_8859-1:1987.0`	■ iso-ir-100
■ `ibm819`	■ `cp819`

To change the character set, follow these steps:

▼ Changing Character Set

1. **From the Virtual Server page, click the Content Handling tab.**
2. **Click the General tab.**
3. **Set the default character set under the Miscellaneous section.**
   If you leave this field blank, the character set is set to NONE.
4. **Click Save.**

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:

▼ To Set the Document Footer

1. **From the virtual server page, click the Content Handling tab.**
2. **Click General sub tab and go to Document Footer section.**
3. **Specify the type of files that you want to include in the footer.**
4. **Specify the date format.**
5. **Type any text you want to appear in the footer.**
   The maximum number of characters for a document footer is 7.065. If you want to include the date the document was last modified, type the string `:LASTMOD:`.
6. **Click Save.**

Note – Using CLI

To set the document footer through CLI, execute the following command.

```
wadm> enable-document-footer --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
--mime-type=text/html --date-format=%B --footer="config1 footer"
```

See CLI Reference, `enable-document-footer(1)`.

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. People can thus create pointers to sensitive files such as confidential documents or system password files.

To restrict symbolic links, follow these steps:

▼ To Restrict Symbolic Links

1. **From the virtual server page, click the Content Handling tab.**
2. **Click the General sub tab.**
3. **Go to the Symbolic Links section under Miscellaneous.**
4. **Choose whether to enable soft and/or hard links and the directory to start from.**
5. **Click Save**

Note – Using CLI

To restrict symbolic links through CLI, execute the following command.

```
wadm> set-symlinks-prop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
allow-soft-links=true allow-hard-links=false directory=/abc
```

See CLI Reference, `set-symlinks-prop(1)`.

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:

▼ To Set Server Parsed HTML

1. **From the virtual server page, click the Content Handling tab.**
2. **Click the General sub tab.**
3. **Under Parsed HTML/SSI Settings, 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 enables HTML files to execute other programs on the server.
4. **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.
5. **Click Save.**

   For more information on using server-parsed HTML, see the Web Server *Developer's Guide.*

Note – Using CLI

To set up server parsed HTML through CLI, execute the following command.

```
wadm> enable-parsed-html --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs1
```

See CLI Reference, `enable-parsed-html(1)`.

Setting Cache Control Directives

Cache-control directives are a way for Web Server to control what information a proxy server caches. 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:

▼ To Set Cache Control Directives

1. **From the virtual server page, click the Content Handling tab.**
2. **Click the General sub tab and go to the Cache Control Directives field under Miscellaneous.**
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 Save.**

Note – Using CLI

To set up cache control directives through CLI, execute the following command.

```
wadm> set-cache-control-directives --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 public=true
private=true must-revalidate=true
```

See CLI Reference, `set-cache-control-directives(1)`.

Configuring the Server for Content Compression

Web Server supports HTTP content compression. Content compression enables you to increase delivery speed to clients and serve higher content volumes without increasing hardware expenses. Content compression reduces content download time, a benefit most apparent to users of dialup and high-traffic connections.

With content compression, Web server sends out compressed data and instructs the browser to decompress the data immediately, thus reducing the amount of data sent and increasing page display speed.

Configuring the Server to Serve Pre-Compressed Content

You can configure the 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 pre-compressed 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 pre-compressed content, perform the following steps:

▼ To Change Pre-compressed Content Settings

1. **From the virtual server page, click the Content Management tab.**
2. **Click the General sub tab.**
3. **Go to the Compression > Precompressed Content section and select from the following options.**

- **Precompressed Content** — Enable/Disable. enables you to instruct the server to serve pre-compressed content for the selected resource.
- **Age Checking** — Specifies whether to check if the compressed version is older than the non-compressed version.

  If selected, then the compressed version, even if it is older than the non-compressed version, will not be checked.
- **Insert Vary Header** — Specifies whether to use a `Vary: Accept-encoding` header.

  If selected, then a `Vary: Accept-encoding` header is always inserted when a compressed version of a file is selected.

  If not selected, then a `Vary: Accept-encoding` header is never inserted.

4 **Click Save.**

Configuring the Server to Compress Content on Demand

You can also configure the server 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 the server to compress content on demand, perform the following steps:

▼ To Compress Content on Demand

1 **From the virtual server page, click the Content Handling tab.**

2 **Click the General sub tab. Go to Compression > Compress Content on Demand.**

3 **Select from the following options:**

- **On—Demand Compression** — Enables/Disables on-demand compression for the selected resource.
- **Insert Vary Header** — Specifies whether to insert a `Vary: Accept-encoding` header.

  If selected, then a `Vary: Accept-encoding` header is always inserted when a compressed version of a file is selected.

  If not selected, then a `Vary: Accept-encoding` header is never inserted.
- **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 Save.**

Note – Using CLI

To enable compression on demand through CLI, execute the following command.

```
wadm> enable-on-demand-compression --user=admin
--password-file=admin.pwd --host=serverhost --port=8989 --config=config1
--vs=config1_vs_1 --insertvaryheader=true
--fragment-size=100 --compression-level=5
```

See CLI Reference, `enable-on-demand-compression(1)`.

Setting Up P3P

- "Configuring Virtual Server's P3P Settings" on page 157

Platform for Privacy Preferences (P3P) enables web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents enable users to receive information about site practices (in both machine- and human-readable formats). For more information, see `http://www.w3.org/P3P/`.

▼ Configuring Virtual Server's P3P Settings

1 **Click the Configurations tab to see the list of available configurations and select the configuration you need.**

2 **Click the Virtual Servers tab to see the available servers and select the virtual server from the list.**

3 **Click the General tab and configure the following settings under P3P section.**

- **Enabled** — Enable P3P for the selected virtual server.
- **Policy URL**— Enter the location of the relevant P3P policy file.
- **Compact Policy** — Compact policies provide hints to user agents (browsers or other P3P applications) to enable the user agent to make quick, synchronous decisions about applying policy. Compact policies are a performance optimization that is meant by the P3P specification to be optional for either user agents or servers.

Note – Using CLI

To enable P3P for the virtual server, execute the following command:

```
wadm> enable-p3p --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --policy-url=http://xyz.com/policyurl
```

See CLI Reference, `enable-p3p(1)`.

◆ ◆ ◆ CHAPTER 10

Web Publishing With WebDAV

Web Server supports WebDAV or Web-based Distributed Authoring and Versioning, a standard in Web-based collaboration. WebDAV is an extension to the HTTP/1.1 protocol that enables clients to perform remote web content authoring operations.

A complete WebDAV transaction involves a WebDAV-enabled server, such as Web Server 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 Web Server to be able to service WebDAV requests.

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.

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 prevent 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 Web Server provides the following features:

- Compliance with RFC 2518 and interoperability with RFC 2518 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` which resides on the server, but a dynamically generated page that displays the current date.

If you were to create a source URI, for example, `/publish/docs`, and map it to the `/docs` directory containing `foo.jsp`, then a request for `/publish/docs/foo.jsp` will be a request for the source code of the `/docs/foo.jsp` JSP page. In this case, the server will serve the page without executing the Java code. The client will 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.

The server supports the following live properties:

- `creationdate`
- `displayname`
- `getcontentlanguage`
- `getcontentlength`
- `getcontenttype`

- gettag
- getlastmodified
- lockdiscovery
- resourcetype
- supportedlock
- executable

Note – The server supports the live property `executable` that enables 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.

The 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 will crawl the entire collection, it can 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.

Enable WebDAV at Instance Level

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 execute the following command in CLI.

```
wadm> enable-webdav --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=test
```

See CLI Reference, `enable-webdav(1)`.

Managing WebDAV Collections

Enabling WebDAV Collection

To enable WebDAV collection, execute the following command:

```
wadm> enable-dav-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

See CLI Reference, `enable-dav-collection(1)`.

Disabling WebDAV Collection

To disable WebDAV collection, execute the following command:

```
wadm> disable-dav-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

See CLI Reference, `disable-dav-collection(1)`.

Adding a WebDAV Collection

To add a WebDAV Collection, execute the following command:

```
wadm> create-dav-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
--source-uri=/dav_config1
```

See CLI Reference, `create-dav-collection(1)`.

Listing WebDAV Collections

To list all WebDAV collections, execute the following command:

```
wadm> list-dav-collections --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1
```

See CLI Reference, `list-dav-collections(1)`.

Removing WebDAV Collection

To remove WebDAV collection, execute the following command:

```
wadm> delete-dav-collection --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

See CLI Reference, `delete-dav-collection(1)`.

Configuring WebDAV Properties

Setting WebDAV Properties

To set WebDAV properties at the server level, execute the following command:

```
wadm> set-webdav-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 acl-max-entries=120
```

See CLI Reference, `set-webdav-prop(1)`.

Viewing WebDAV Properties

To view WebDAV properties at the server level, execute the following command:

```
wadm> get-webdav-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1
```

See CLI Reference, `get-webdav-prop(1)`.

Setting WebDAV Collection Properties

To set WebDAV collection properties, execute the following command:

```
wadm> set-dav-collection-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1 min-lock-timeout=1
```

See CLI Reference, `set-dav-collection-prop(1)`.

Viewing WebDAV Collection Properties

To view WebDAV collection properties, execute the following command:

```
wadm> get-dav-collection-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 -config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

See CLI Reference, `get-dav-collection-prop(1)`.

Modifying WebDAV Parameters

Some common WebDAV properties are listed in the following table:

TABLE 10–1 WebDAV Parameters

Parameter	Description
Lock Database Path	Specify the directory in which the locking database will be maintained.
Minimum Lock Time-out	Specify the minimum lifetime of a lock in seconds. A value of **-1** implies that the lock never expires. This value indicates the amount of time that an element will be locked before the lock is automatically removed.
Maximum Request Size	Specify the maximum size of the XML request body. You should configure this value to prevent possible denial of service attacks. The default value is **8192 (8K).**
Maximum Expand Property Depth	Specify the depth of the Expand Property. **0** applies only to the specified resource. This is the default value. **1** applies to the specified resource and the next level. infinity applies to the specified resource and all resources it contains. Also prevent excessive memory consumption by restricting the size of this parameter.
Default Owner	Default owner for the collection.
URI	Existing root URI on which WebDAV will be enabled.
Maximum PROPFIND Depth	Maximum depth of PROPFIND requests send to collections.
Lock Database Update Interval	Interval at which WebDAV lock databases are synced to disk. Use 0 to disable caching of WebDAV lock information.
Authentication Database	The ACL authentication database to use.
Authentication Method	The authentication method to use. Default authentication method is Basic.

TABLE 10–1 WebDAV Parameters *(Continued)*

Parameter	Description
Authentication Prompt Text	The prompt to display to clients when requesting authentication.
DAV ACL Database	
Maximum Entries	Maximum number of ACEs to allow on a single resource. 0–2147.0483647.0. Specify —1 for no limit.
Maximum Size	Maximum size of the memory representation of the WebDAV ACL database for a collection. 0–2147.0483647.0. Specify —1 for no limit.
Update Interval	Interval at which WebDAV ACL databases are synced to disk. 0.001–3600 seconds. Specify 0 to disable caching of WebDAV ACL lists.
DAV Property Database	
Maximum Size	Maximum size of WebDAV property database files. 0–2147.0483647.0. Specify —1 for no limit.
Update Interval	Interval at which WebDAV property databases are synced to disk. 0.01–3600 seconds. Specify 0 to disable caching of WebDAV properties.

Disabling WebDAV at Server Level

To disable WebDAV at server level, execute the following command:

```
wadm> disable-webdav --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1
```

See CLI Reference, `disable-webdav(1)`.

Managing WebDAV Authentication Databases

From the administration console, click **WebDAV** tab from the selected configuration to edit WebDAV authentication database settings. The following table provides a short description of each field in the page:

TABLE 10–2 WebDAV Authentication Database Properties

Property	Description
Authentication Database	*Authentication Database* lets you select a database the server will use to authenticate users. The default is **keyfile**
Authentication Method	■ **Basic** — uses the HTTP Basic method to obtain authentication information from the client. The username and password are only encrypted over the network if SSL 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 an authentication mechanism that provides a way for a browser to authenticate based on username and password without sending the username and password as clear text. The browser uses the *MD5* algorithm to create a digest value using the user's password and some information provided by the Web Server. Note that in order to use Digest the underlying auth-db must support digest as well. This means either a File auth-db using digestfile or an LDAP auth-db only if the Digest Authentication Plug-in has been installed ■ **Other** — uses a custom method created using the access control API.
Authentication Prompt Text	**Prompt for Authentication** option enables 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 browser, the user will see about the first 40 characters of the prompt. Web browsers typically 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.

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, the server 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, the server defines a 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.

Locking and Unlocking Resources

The server enables 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.

The server 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-1157.08.

The server recognizes two types of locking mechanisms:

- Exclusive Locks.
- Shared Locks.

Exclusive Locks

An exclusive lock is a lock that grants resource access to 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 will be required to remove the exclusive lock.

Shared Locks

A shared lock enables 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.

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` enables 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 the server handles locking requests:

TABLE 10–3 How Sun Java System 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)

TABLE 10–3 How Sun Java System Web Server handles locking requests *(Continued)*

If Timeout header value in Request is set to:	The resource is:
`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`.

Note – Using CLI

To set lock expiry through CLI, execute the following command:

```
wadm> expire-lock  --user=admin --password-file=admin.pwd
--host=serverhost--port=8989 --config=config1 --vs=config1
--collection-uri=/dav1 --lock-uri=/dav1/file.html
--opaque-token=opaquelocktoken
```

See CLI Reference, `expire-lock(1)`.

In the above example opaque-token specifies the ID of the lock that you want to set to expire.

To display existing locks through CLI, execute the following command:

```
wadm> list-locks --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs config1 --collection-uri=/dav1
```

See CLI Reference, `list-locks(1)`.

◆ ◆ ◆ CHAPTER 11

Working With Java and Web Applications

This chapter describes the procedures for editing Java settings for a virtual server. You can edit Java settings from the administration console or the `wadm` command line tool. This chapter also describes various Java resources that can be configured and deployed in the server.

Configure Java to Work With Sun Java System Web Server

This section lets you enable Java and set Java Home variable for the selected configuration.

▼ Enabling Java for Your Configuration

1. **Click the Configuration tab to see the list of available configurations and select the configuration you need.**
2. **Click Java > General tab.**
3. **Select Enable Java check box.**

   Turn the Java support on or off for the configuration. Enabling Java allows the server to process Java applications.

4 **Set Java Home by specifying the location of Java SE.**

Specify the absolute path or path relative to the server's `config` directory.

5 **Set Stick Attach by specifying whether the server attaches each HTTP request processing thread to the JVM only once.**

Otherwise the server attaches/detaches the HTTP request processing thread on each request.

Note – Using CLI

To enable Java for a configuration, execute the following command.

```
wadm> enable-java --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1
```

See CLI Reference, `enable-java(1)`.

Setting Up Java Class Path

This section enables you to add JVM class path for the selected configuration.

▼ To Set Up Java Class Path

1 **Click the Configuration tab to see the list of available configurations and select the configuration you need.**

2 **Click Java > Path Settings tab.**

Edit the following parameters:

- **Ignore Environment Class Path** — Enabled by default.
- **Class Path Prefix** — Prefix for the system class path. You should only prefix the system class path if you wish to override system classes, such as the XML parser classes. **Use this with caution**.
- **Server Class Path** — Class path containing server classes. Read-Only list.
- **Class Path Suffix** — Append to server class path.
- **Native Library Path Prefix** — Prefix for the operating system native library path.
- **Bytecode Preprocessor Class** — Fully qualified name of a class that implements `com.sun.appserv.BytecodePreprocessor`. A typical way to perform runtime class instrumentation is through the preprocessing mechanism, whereby profiling and monitoring tools use a class preprocessor to insert instrumentation code at the required places in the Java classes just before they are loaded by the JVM. Toward that end, the class preprocessor works in conjunction with the class loader.

Configuring Your JVM

To set JVM command-line options in the Administration interface, perform the following tasks:

▼ To Configure Your JVM

1 **Click the Configuration tab and select the configuration from the configuration list.**

2 **Click Java > JVM Settings tab.**

Configure the settings for your JVM.

Adding a JVM Option

You can add or delete command line JVM options by specifying the values here.

Click Add JVM Option to add a JVM option.

Some examples for JVM options are:`-Djava.security.auth.login.config=login.conf`, `-Djava.util.logging.manager=com.iplanet.ias.server.logging.ServerLogManager` and `-Xms128m -Xmx256m`

Note – Using CLI

To add JVM options through CLI, execute the following command.

```
wadm> create-jvm-options --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 -Dhttp.proxyHost=proxyhost.com -Dhttp.proxyPort=8080
```

See CLI Reference, `create-jvm-options(1)`.

Adding JVM Profilers

JVM Profiler helps you diagnose and resolve performance problems, memory leaks, multi-threading problems and system resource usage problems in your Java applications to ensure the highest level of stability and scalability for your applications.

To Add a JVM Profiler

1. **Click Configurations tab to see the list of available configurations and select the configuration you need.**
2. **Click Java > JVM Settings tab.**
3. **Under the Profilers section, click New.**
4. **Provide values for the following parameters:**
   - **Name** — Provide a short name for the new JVM Profiler.
   - **Enabled** — Determines if the profiler is enabled at runtime.
   - **Class path** — Provide a valid class path for the profiler. (Optional).
   - **Native library path** — Provide a valid native library path. (Optional).
   - **JVM Options** — You can specify additional JVM options for the CLI.

Note – Using CLI

To add a JVM profiler through CLI, execute the following command.

```
wadm> create-jvm-profiler --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
```

See CLI Reference, `create-jvm-profiler(1)`.

Enabling Java Debugging for Your Server

The JVM can be started in debug mode and can be attached to a JPDA (Java Platform Debugger Architecture) debugger. When you enable debugging, you enable both local and remote debugging.

Sun Java System Web Server's debugging is based on JPDA software. To enable debugging, perform the following tasks.

Enable JVM Debugging

1. **Click the Configurations tab to see the list of available configurations and select the configuration you need.**
2. **Click Java > JVM Settings tab.**
3. **Under Debug Java Settings, select the Enable Debug checkbox.**

4 **Provide JVM options as necessary by clicking the New button.**

The default JPDA options are as follows:

```
-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=7896
```

If you substitute `suspend=y`, the JVM starts in suspended mode and stays suspended until a debugger attaches to it. This is helpful if you want to start debugging as soon as the JVM starts. To specify the port to use when attaching the JVM to a debugger, specify `address=port_number`. Check out the JPDA documentation for a list of debugging options.

Deploying Java Web Applications

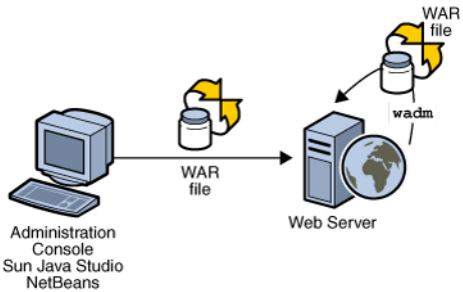
Adding a Web Application

You can deploy a web application to any existing virtual server.

▼ To deploy a web application

Before You Begin

- Identify a virtual server where you will need to deploy the web application.
- Be sure you have either the web application archive (.war file) or know the web application path in the server.



Web applications can be deployed through `wadm`, Administration Console and other supported IDEs.

1 **To deploy a web application, click Server Configuration and then click the Virtual Servers tab.**

2 **Select the virtual server in which you will need to deploy the web application.**

3 **Click the Web Applications tab > New button.**

4 **Specify the web application package.**

If you need to upload a web application archive, click the Browse button and select the archive. Optionally, you can also specify a web application archive located in the server.

5 **Specify the URI for your web application. The URL will be the applications context root and is relative to the server host.**

6 **Provide a short description about the web application.**

7 **Enable/Disable JSP Pre-compilation.**

Enabling this directive will allow all the JSPs present in the web application to be pre-compiled to improve performance.

8 **Enable the application.**

When a web application state is set to be Disabled, it will not be available on request. However you can toggle this option anytime without redeploying the application to the instances.

9 **Deploy the application.**

Click Deploy to deploy the web application.

You can access the application with the context root specified. E.g.
`http://<your-server>:<port>/<URI>`

Note – Using CLI

```
wadm> add-webapp --user=admin --password-file=admin.passwd --host=localhost
 --port=8888 --config=config1 --vs=HOSTNAME --uri=/hello /home/test/hello.war
```

See CLI Reference, `add-webapp(1)`.

Deploying a Web Application Directory

A directory on the administration server host machine can be deployed to a configuration using the `–file-on-server` option. Execute the following command:

```
wadm> add-webapp --user=admin-user --password-file=admin.passwd
--port=8989 --vs=vs1 --config=config1 --file-on-server
--uri=/mywebapp /space/tmp/mywebapp
```

Pre-compiling JSPs During Deployment

To pre-compile JSPs in a web application while deploying the web application, execute the command with `–precompilejsp` option as given below:

```
wadm> add-webapp --user=admin-user --password-file=admin.passwd
--port=8989 --vs=vs1 --config=config1 --file-on-server --uri=/mywebapp
--precompilejsp mywebapp.war
```

Configuring Your Servlet Container

This section describes the procedure for configuring the servlet container.

▼ To Set Up Servlet Container

1. **Click the Configurations tab to see the list of available configurations and select the configuration you need.**
2. **Click Java > Servlet Container.**

Servlet Container Global Parameters

The following table describes the parameters available on the servlet container page.

TABLE 11–1 Servlet Container Parameters

Parameter	Description
Log Level	Log verbosity for the servlet container. The values can be finest (most verbose), finer, fine, info, warning, failure, config, security, or catastrophe (least verbose).
Dynamic Reload Interval	Defines the time period after which the server checks deployed web applications for modifications. The value range is 1 to 60, or –1 if dynamic reloading should be disabled.
Anonymous Role	Name of the default, or anonymous, role assigned to all principals. The default role is ANYONE.
Servlet Pool Size	Number of servlet instances to instantiate per `SingleThreadedServlet`. The range value is 1 to 4096.
Dispatcher Max Depth	Maximum depth for the servlet container allowing nested request dispatches. The range of values can be between 0 and 2147.0483647.0. The default value is 20.
Allow Cross Context	Tells whether request dispatchers are allowed to dispatch to another context. The default value is false.

TABLE 11–1 Servlet Container Parameters *(Continued)*

Parameter	Description
Encode Cookies	Indicates whether the servlet container encodes cookie values. The default value is true.
Display Exception	Displays an exception on the browser. This option is useful only in development environment. Disable this option in production environment.
Decode Cookies	The servlet container decodes the plus character in cookie value to space.
Reuse Session IDs	Indicates whether any existing session ID number is reused when creating a new session for that client. The default value is false.
Secure Session Cookie	Dynamic/True/False. This parameter controls under what conditions the JSESSIONID cookie is marked secure. Use dynamic (default) to mark the cookie secure only when the request was received on a secure connection (HTTPS). Select True to always mark it secure and false to never mark it secure.

Configuring Server Lifecycle Modules

Java Server Lifecycle Modules are Java classes that listen for server lifecycle events in order to perform certain tasks.

The server supports running short or long duration Java-based tasks within the web server environment. These tasks are automatically initiated upon server startup and are notified upon server shutdown. So now you can link tasks such as instantiating singletons and RMI servers.

A brief description of the Server's lifecycle is given below.

Introduction to Server Lifecycle

- **init** — This phase includes reading configuration, initializing built-in subsystems; naming, security and logging services; and creating the web container.
- **startup** — This phase includes loading and initializing deployed applications
- **service** — The server is ready to service requests
- **shutdown** — This phase stops and destroys loaded applications. The system is preparing to shutdown.
- **termination** — This phase terminates the built-in subsystems and server runtime environment. There won't be any more activity after this phase.
- **reconfig** — The transient server state in which a server thread is dynamically reconfiguring (while the server is in the service state). This phase can occur several times during the life of the server.

▼ To Add a Lifecycle Module

1. **Click the Configuration tab and select the configuration you need.**
2. **Click Java > Lifecycle Modules tab.**
3. **Click New.**

   Provide values for the following parameters:

   - **Name** — Provide a valid unique name for the new lifecycle module.
   - **Enabled** — If you want to enable this lifecycle module, use this option.
   - **Class Name** — Fully qualified Java class name. The class should implement `com.sun.appserv.server.LifecycleListener` interface. For more information on using this interface, refer to the *Developer's Guide*.
   - **Class Path** — Optional. You can specify a class path to the listener class.
   - **Load Order** — Greater than 100. Order of loading the lifecycle event listeners, in the numerical order. It is recommended to choose a load-order that is greater than or equal to 100 to avoid conflicts with internal lifecycle modules.
   - **On Load Failure** — If this option is enabled, the server does not treat exceptions thrown from the listener classes as fatal thus it continues with the normal startup. Disabled by default.
   - **Description** — Provide a short description about the lifecycle module.
   - **Properties** — Properties can be used to pass arguments to a Java Lifecycle Module. To add a new property, click Add Property button and enter text for name, value and description.

Caution – The server lifecycle listener classes are invoked synchronously from the main server thread and hence extra precaution must be taken to ensure that the listener classes do not block the server. The listener classes may create threads if appropriate but they must be stopped during the shutdown/termination phases.

▼ To Delete a Lifecycle Module

1. **Click the Configuration tab to view the list of configurations and select the configuration you need.**
2. **Click Java > Lifecycle Modules tab.**
3. **Select the lifecycle module and click Delete Lifecycle Module.**

Note – Using CLI

The following example depicts how to create a Java Lifecycle Module named `myLifecycleModule` for the configuration `test`, implemented by the class `com.MyLifecycleModule`.

```
wadm> create-lifecycle-module --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--class=com.sun.webserver.tests.LifecycleClass LifecycleTest
```

See CLI Reference, `create-lifecycle-module(1)`.

To list Java Lifecycle Modules execute the following command:

```
wadm> list-lifecycle-modules --config=test
```

See CLI Reference, `list-lifecycle-modules(1)`.

To add properties to Java Lifecycle Modules, execute the following command:

```
wadm> create-lifecycle-module-userprop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --module=LifecycleTest info=Testing
```

See CLI Reference, `create-lifecycle-module-userprop(1)`.

To modify Java Lifecycle Module properties, execute the following command:

```
wadm> set-lifecycle-module-prop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --module=LifecycleTest
class-path=/space
```

See CLI Reference, `set-lifecycle-module-prop(1)`

Integrating Service Management Facility for the Java™ Platform with Web Server

Service Management Facility for the Java Platform is a new feature in Solaris 10 that creates a unified model for services and service management on each Solaris system.

Managing Service Management Facility on Web Server Instances

The following `svcadm` commands helps to manage Service Management Facility on Web Server.

Note – During installation of Web Server, you can choose to install service for Administration Server.

- `svcadm enable <service-name>:<instance-name>` - Starts the instance.
- `svcadm disable <service-name>:<instance-name>`- Stops the instance.
- `svcadm refresh <service-name>:<instance-name>` - Restarts the instance.
- `svcadm clear <service-name>:<instance-name>` — Clears the state of the instance. You can also use the `svcadm clear` command to change the service state from maintenance to stop, when the service turns to maintenance state.

You can create a service while creating an instance. Use the following command to create service while creating an instance:

```
wadm>create-instance <connect_options> --echo --no-prompt --verbose --force
--config=<config_name> name --create-service (nodehost)+
```

Use the following command to create a service in an existing instance:

```
wadm>create-service –config=<config-name> node host
```

To learn more about creating a instance through CLI, see`create-instance(1)` see

Service Manifest for Web Server

A service is usually defined by a service manifest, an XML file which describes a service and any instances associated with that service. The service manifest is imported into the repository by using the `svccfg import` command. Service Management Facility requires all manifest file for services to be in the following location `/var/svc/manifest`.

Note – Use `delete-service` command to delete the service.

The following is the sample manifest file for Web Server:

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
```

```
<!-- Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
     Use is subject to license terms. -->

<service_bundle type='manifest' name='webserver7'>
    <service name='network/http' type='service' version='1'>
        <dependency name='filesystem' grouping='require_all' restart_on='none' type='service'>
            <service_fmri value='svc:/system/filesystem/local'/>
        </dependency>
    <instance name='admin-server' enabled='false'>
            <property_group name='start' type='method'>
                <propval name='exec' type='astring' value='/var/opt/SUWwbsvr7/admin-server/bin/startserv'/>
                <propval name='instanceRoot' type='astring' value='/var/opt/SUWwbsvr7'/>
                <propval name='timeout' type='astring' value='300'/>
            </property_group>
            <property_group name='stop' type='method'>
                <propval name='exec' type='astring' value='/var/opt/SUWwbsvr7/admin-server/bin/stopserv'/>
                <propval name='timeout' type='astring' value='300'/>
            </property_group>
            <property_group name='refresh' type='method'>
                <propval name='exec' type='astring' value='/var/opt/SUWwbsvr7/admin-server/bin/restartserv'/>
                <propval name='timeout' type='astring' value='300'/>
            </property_group>
            <property_group name='startd' type='framework'>
                <propval name='ignore_error' type='astring' value='core,signal'/>
            </property_group>
        </instance>
    <instance name='https-mycompany.com' enabled='false'>
            <property_group name='start' type='method'>
                <propval name='exec' type='astring' value='/var/opt/SUWwbsvr7/https-mycompany.com/bin/startserv'/>
                <propval name='instanceRoot' type='astring' value='/var/opt/SUWwbsvr7'/>
                <propval name='timeout' type='astring' value='300'/>
            </property_group>
            <property_group name='stop' type='method'>
                <propval name='exec' type='astring' value='/var/opt/SUWwbsvr7/https-mycompany.com/bin/stopserv'/>
                <propval name='timeout' type='astring' value='300'/>
            </property_group>
            <property_group name='refresh' type='method'>
                <propval name='exec' type='astring' value='/var/opt/SUWwbsvr7/https-mycompany.com/bin/restartserv'/>
                <propval name='timeout' type='astring' value='300'/>
            </property_group>
            <property_group name='startd' type='framework'>
                <propval name='ignore_error' type='astring' value='core,signal'/>
            </property_group>
        </instance>
        <stability value='Evolving' />
        <template>
            <common_name>
            <loctext xml:lang='C'>Sun Java System Web Server 7</loctext>
```

```
            </common_name>
          </template>
      </service>
</service_bundle>
```

Service Log

The service log file is located in the following directory `/var/svc/log`. Service log file entries contain information about the attempted action, the outcome of the action, and the cause of failure if applicable. The service logs are located as follows `/var/svc/log/network-http:admin-server.log`.

Configuring Java 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 Java EE platform exposes such resources to the applications through the Java Naming and Directory Interface (JNDI) service.

The Sun Java System Web Server enables you to create and manage the following Java EE resources:

- JDBC Datasources.
- JDBC Connection Pools.
- Java Mail Sessions.
- Custom Resources.
- External JNDI Resources.

Configuring JDBC Resources

A JDBC Datasource is a Java EE resource that you can create and manage using the Sun Java System 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 Java EE 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 Java EE resource that can be created and managed through the Sun Java System 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.

Adding a JDBC Resource

To add a JDBC resource through CLI, execute the following command.

```
wadm> create-jdbc-resource --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --datasource-class=oracle.jdbc.pool.OracleDataSource jdbc
```

See CLI Reference, `create-jdbc-resource(1)`.

In the previous example, `com.pointbase.jdbc.jdbcDataSource` represents the JDBC driver class.

For a list of supported JDBC drivers, see "JDBC Drivers Known to Work With the Sun Java System Web Server" on page 186.

JDBC Drivers Known to Work With the Sun Java System Web Server

The following table provides a list of common JDBC drivers and their properties. These drivers need to be configured while adding a new JDBC resource. See "Adding a new JDBC Resource" on page 189.

TABLE 11–2 List of common and JDBC drivers

Driver	Class Name	Properties
Oracle driver	`oracle.jdbc.pool.OracleDataSource`	■ url ■ user ■ password
Sun Java System JDBC driver for Oracle	`com.sun.sql.jdbcx.oracle.OracleDataSource`	■ serverName ■ portNumber ■ user ■ password ■ SID

TABLE 11–2 List of common and JDBC drivers *(Continued)*

Driver	Class Name	Properties
DB2 IBM driver	`com.ibm.db2.jdbc.DB2DataSource`	■ serverName ■ databaseName ■ portNumber ■ user ■ password ■ driverType
Sun Java System JDBC driver for DB2	`com.sun.sql.jdbcx.db2.DB2DataSource`	■ databaseName ■ locationName ■ packageName ■ password ■ portNumber ■ serverName ■ user
MS SQLServer driver	`com.ddtek.jdbcx.sqlserver.SQLServerDataSource`	■ databaseName ■ password ■ user ■ serverName ■ portNumber
Sun Java System JDBC driver for MS	`com.sun.sql.jdbcx.sqlserver.SQLServerDataSource`	■ databaseName ■ password ■ user ■ serverName ■ portNumber
Sybase driver	`com.sybase.jdbcx.SybDataSource`	■ databaseName ■ password ■ portNumber ■ serverName ■ user
Sun Java System JDBC driver for Sybase	`com.sun.sql.jdbcx.sybase.SybaseDataSource`	■ databaseName ■ password ■ user ■ portNumber ■ serverName

TABLE 11–2 List of common and JDBC drivers *(Continued)*

Driver	Class Name	Properties
MySQL driver	`com.mysql.jdbc.jdbc2.optional.MysqlDataSource`	■ serverName ■ port ■ databaseName ■ user ■ password
Informix driver	`com.informix.jdbcx.IfxDataSource`	■ portNumber ■ databaseName ■ IfxIFXHOST (The IP address or the host name of the computer running the Informix database) ■ serverName ■ user ■ password
Sun Java System JDBC driver for Informix	`com.sun.sql.jdbcx.informix.InformixDataSource`	■ databaseName ■ informixServer (The name of the Informix database server to which you want to connect) ■ password ■ portNumber ■ severName
PostgreSQL driver	`org.postgresql.ds.PGSimpleDataSource`	■ serverName ■ databaseName ■ portNumber ■ user ■ password
Apache Derby driver	`org.apache.derby.jdbc.EmbeddedDataSource`	■ databaseName ■ user ■ password

Note – In the previously mentioned list, all of the Sun Java System JDBC drivers are shipped with the Web Server. For other drivers, check with the driver vendor documentation for the latest versions of these drivers and the class names. The information provided in the previously mentioned list may not be the latest driver information.

Managing JDBC Resources

▼ Adding a new JDBC Resource

1. **Click the Configuration tab and select the configuration from the configuration list.**
2. **Click Java > Resources tab.**
3. **Under JDBC Resources, click New.**
4. **Select the Driver Vendor.**

   Specify a unique value for the JNDI name and select the JDBC driver vendor from the available list.
5. **Provide JDBC Resource Properties.**

   Based on the JDBC driver vendor selection in the previous step, the class name for the driver and the JDBC resource properties are automatically populated.
6. **Review.**

   View the summary and click Finish to create the new JDBC resource.

Managing JDBC Connection Pools

Configuring JDBC Connection Pool

In Web Server 7.0, JDBC Connection Pools are configured through JDBC resource elements. The simplest connection pool can be configured by following the steps listed below. In this example, the connection pool will use the Oracle JDBC driver.

▼ To Create a JDBC Connection Pool

1 **Start wadm.**

2 **Create a JDBC Resource with the basic configuration.**

Other attributes are available to fine tune the connection pool. Refer to the Manual Pages for more attributes and examples.

```
wadm> create-jdbc-resource --config=test
--datasourceclass=oracle.jdbc.pool.OracleDataSource jdbc/MyPool
```

3 **Configure Vendor Specific Properties.**

Properties are used to configure the driver's vendor specific properties. In the example below the properties `url`, `user` and `password` are added to the JDBC resource.

```
wadm> add-jdbc-resource-userprop --config=test --jndi-name=jdbc/MyPool
url=jdbc:oracle:thin:@hostname:1521:MYSID user=myuser password=mypassword
```

4 **Enable Connection Validation.**

Connection validation can be enabled for the pool. If this option is used, connections will be validated before they are passed to the application. This enables 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.

```
wadm> set-jdbc-resource-prop --config=test --jndi-name=jdbc/MyPool
connection-validation-table-name=test connection-validation=table
```

5 **Change Default Pool Settings.**

In this example, change the maximum number of connections.

```
wadm> set-jdbc-resource-prop --config=test --jndi-name=jdbc/MyPool
max-connections=100
```

6 **Deploy the Configuration.**

```
wadm> deploy-config test
```

7 **Provide the Jar Files Containing the JDBC driver.**

The server needs to be provided with the classes that implement the driver. This can be done in two ways:

- Copy the driver's jar file into the server instance lib directory. This is the simplest way, as the jar files included in the instance lib directory will be automatically loaded and available to the server.
- Modify the JVM's *class-path-suffix* to include the JDBC driver's jar file.

```
wadm> set-jvm-prop  --config=test class-path-suffix=/export/home/lib/classes12.jar
```

8 Usage in Web Applications.

- Modifying `WEB-INF/web.xml`.

```
<web-app>
...
  <resource-ref>
    <description>JDBC Connection Pool</description>
    <res-ref-name>jdbc/myJdbc</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
...
</web-app>
```

- Modifying `WEB-INF/sun-web.xml`.

```
<sun-web-app>
...
  <resource-ref>
    <res-ref-name>jdbc/myJdbc</res-ref-name>
    <jndi-name>jdbc/MyPool</jndi-name>
  </resource-ref>
...
</sun-web-app>
```

- Using the Connection Pool.

```
Context initContext = new InitialContext();
    Context webContext = (Context)context.lookup("java:/comp/env");

    DataSource ds = (DataSource) webContext.lookup("jdbc/myJdbc");
    Connection dbCon = ds.getConnection();
```

Registering Custom Resources

You can register a custom resource with the instance by performing this task.

▼ To Add a Custom Resource

1 Click Configurations tab and select the configuration from the list.

2 Click Java > Resources tab.

3 Under Custom Resource, click New.

Properties for Custom Resources

The following table describes the properties available for creating a custom resource.

TABLE 11–3 Custom Resources Properties

Property	Description
JNDI Name	Provides a unique JNDI name for the custom resource.
Enabled	Determines if this custom resource is enabled at runtime.
Resource Type	Fully qualified type of resource.
Factory Class	Class that instantiates resources of this type. The fully qualified name of the user-written factory class that implements the `javax.naming.spi.ObjectFactory`.
Description	Provide a short description for the custom resource.
Properties	Provides CLI properties. Click Add Property to use.

Note – Using CLI

To create a custom resource through CLI, execute the following command:

```
wadm> create-custom-resource --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --res-type=samples.jndi.customResource.MyBean
--factory-class=samples.jndi.customResource.MyCustomConnectionFactory custom
```

See CLI Reference, `create-custom-resource(1)`.

Working With External JNDI Resources

Creating External JNDI Resources

This option lets you create an external Java Naming and Directory Interface (JNDI) resource. You need a JNDI resource to access resources stored in an external JNDI repository.

▼ To Add an External JNDI Resource

1. **Click the Configuration tab and select the configuration from the list.**
2. **Click Java > Resources tab.**
3. **Under External JNDI, click New.**

Properties for External JNDI Resources

The following table describes the properties available when adding a new external JNDI resource.

TABLE 11–4 External JNDI Resources Properties

Property	Description
JNDI Name	Provides a unique name for the new external JNDI resource.
Enabled	Determines if this external JNDI resource is enabled at runtime.
External JNDI Name	Name of the external JNDI resource.
Resource Type	Fully qualified type of resource.
Factory Class	Class that instantiates resources of this type.
Description	Provides a short description for the external JNDI resource.
Properties	Optionally provides CLI properties. Enabled by clicking the Add Property button.

Note – Using CLI

To create an external JNDI resource through CLI, execute the following command:

```
wadm> create-external-jndi-resource --user=admin
--password-file=admin.pwd --host=serverhost --port=8989 --config=config1
--res-type=org.apache.naming.resources.Resource
--factory-class=samples.jndi.externalResource.MyExternalConnectionFactory
--jndilookupname=index.html external-jndi
```

See CLI Reference, `create-external-jndi-resource(1)`.

Configuring Mail Resources

JMS destinations are Java EE resources that can be created and managed through the Sun Java System Web Server.

Many internet applications require the ability to send email notifications. The Java EE platform includes the JavaMail API along with a JavaMail service provider that enables an application component to send internet mail.

▼ To Add a Mail Resource

1 **Click the Configuration tab to view the list of configurations and select the configuration you need.**

2 **Click Java > Resources tab.**

3 **Under Mail Resource, click New.**

Properties for Mail Resource

The following table describes the properties available while adding a new mail resource.

TABLE 11–5 Mail Resource Properties

Property	Description
JNDI Name	Provides a unique name for the new mail resource.
Enabled	Determines if this mail resource is enabled at runtime.
User	Valid user name registered in the mail server.
From	Email address from which the server sends mail.
Host	Host name/IP address of the mail server.
Store Protocol	Protocol used to retrieve messages.
Store Protocol Class	Storage service provider implementation for store-protocol. Fully qualified class name of a class that implements store-protocol. The default class is `com.sun.mail.imap.IMAPStore`.
Transport Protocol	Protocol used to send messages.
Transport Protocol Class	Transport service provider implementation for transport-protocol. Fully qualified class name of a class that implements transport-protocol. The default class is `com.sun.mail.smtp.SMTPTransport`.

Note – Using CLI

To create a mail resource, execute the following command:

```
wadm> create-mail-resource --config=test --server-host=localhost
--mail-user=nobody --from=xyz@foo.com mail/Session
```

See CLI Reference, `create-mail-resource(1)`.

Configuring SOAP Authentication Providers

The Java Authentication Service Provider Interface for Containers specification defines a standard service provider interface by which authentication mechanism providers may be integrated with containers. You can use the Administration Console to add a new SOAP authentication provider.

▼ To Add a SOAP Authentication Provider

1. **Click Configurations tab and select the configuration you need.**
2. **Click Java > Web Services tab.**
3. **Under SOAP Authentication Provider, click New.**

SOAP Authentication Provider Parameters

The following table describes the parameters available on the new SOAP authentication provider page.

TABLE 11–6 SOAP Authentication Provider Parameters

Parameter	Description
Name	Enter a short name for the new SOAP authentication provider.
Class Name	The class that implements the provider. Fully qualified class name of a class that implements `javax.security.auth.XXX`
Request Authentication Source	This attribute defines a requirement for message layer sender authentication such as username/password or content authentication such as digital signature to be applied to request messages. The value (auth-policy) may be sender or content. When this argument is not specified, source authentication of the request is not required.
Request Authentication Recipient	This attribute defines a requirement for message layer authentication of the receiver of a message to its sender, for example, by XML encryption. The values can be before-content or after-content.
Response Authentication Source	This attribute defines a requirement for message layer sender authentication such as username/password or content authentication such as digital signature to be applied to response messages. The value (auth-policy) may be sender or content. When this argument is not specified, source authentication of the response is not required

TABLE 11–6 SOAP Authentication Provider Parameters *(Continued)*

Parameter	Description
Response Authentication Recipient	This attribute defines a requirement for message layer authentication of the receiver of the response message to its sender, for example, by XML encryption.
Properties	Provides other CLI properties by clicking the Add Property button.

Note – Using CLI

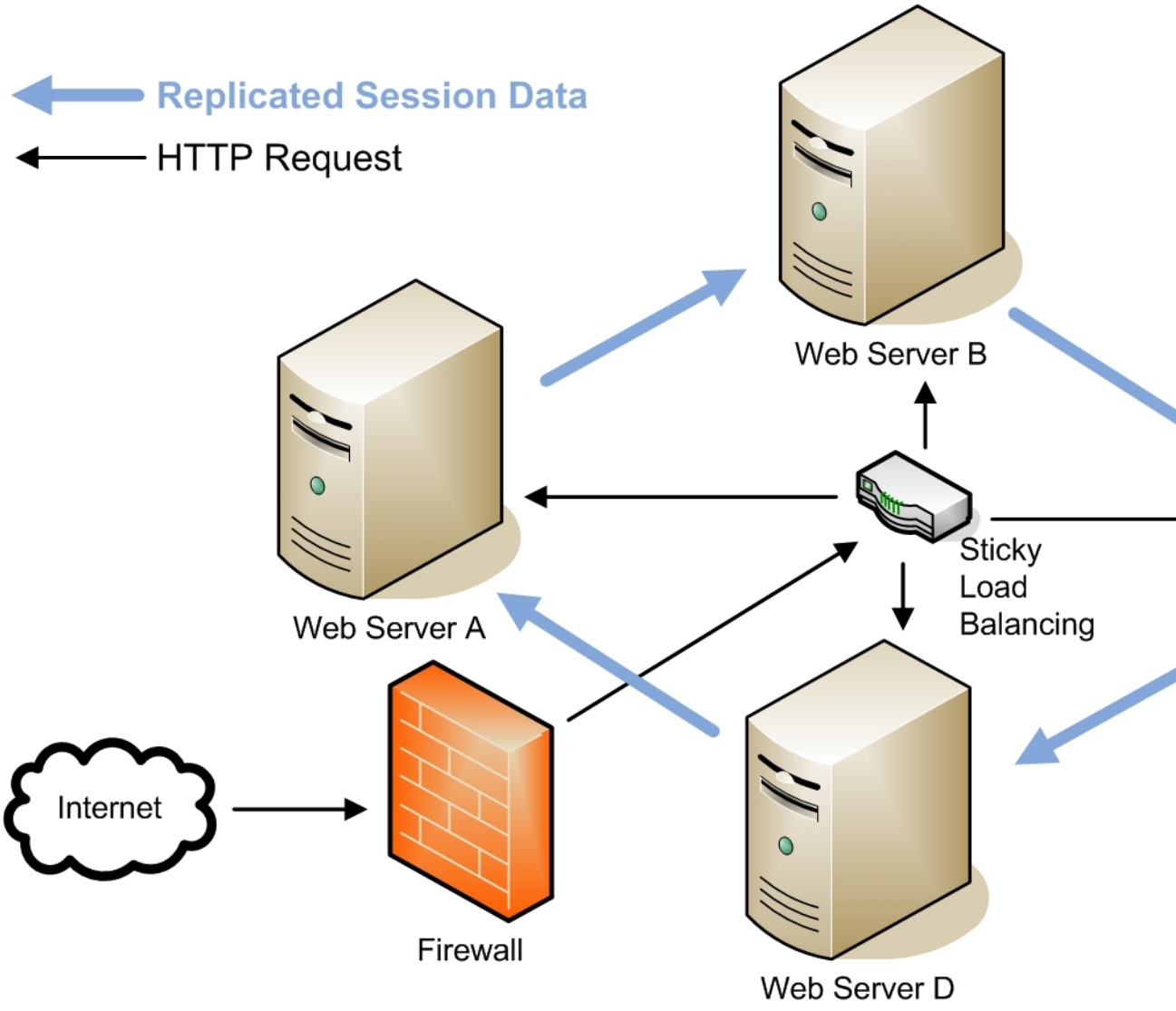
To add a SOAP authentication provider using CLI, execute the following command.

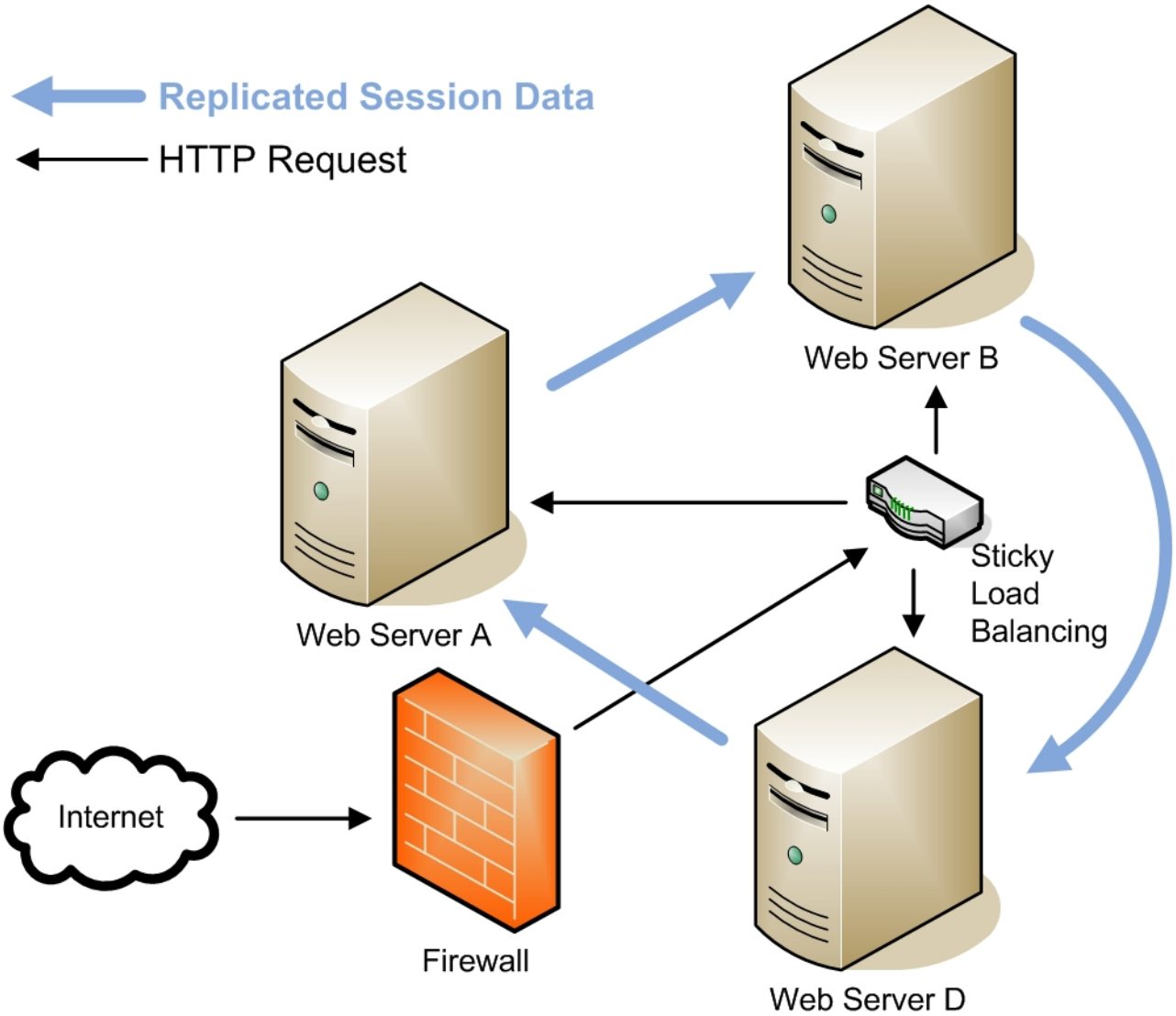
```
wadm> create-soap-auth-provider --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--class-name=javax.security.auth.soapauthprovider soap-auth
```

See CLI Reference, `create-soap-auth-provider(1)`.

Configuring Session Replication

Web Server supports session replication that provides high availability to web applications. Session replication achieves this by replicating HTTP sessions from one instance to another server instance of the same cluster. So, each HTTP session has a backup copy on a remote instance. In the event of a failure which renders one instance in the cluster unavailable, the cluster still maintains session continuity.


Replicated Session Data
HTTP Request
Web Server B
Web Server A
Sticky
Load
Balancing
Internet
Firewall
Web Server D



The above figures depicts a typical scenario when session replication happens between four nodes with a reverse proxy set up. Note that the session data gets replicated from Web Server B to Web Server D when Web Server C goes offline.

Setting Up Session Replication

This section describes the procedure for setting up session replication properties for a selected configuration.

▼ To Set Up Session Replication

1. **Click Configurations tab to see the configuration list and select the configuration you need.**
2. **Click Java > Session Replication.**

Modifying Session Replication Parameters

The following table describes the parameters available on the session replication page.

TABLE 11–7 Session Replication Parameters

Parameter	Description
Port	Port number where the Administration server is listening. The default port is 8888.
Enabled	Enable session replication for the selected configuration.
Encrypted	Whether session data is encrypted prior to replication. The default value is false.
Cipher	The cipher suite (algorithm, mode, padding) the cluster members uses to replicate session data.
Getatrribute triggers replication	Whether a call to the `HttpSession.getAttribute` method should cause a session to be backed up. The default value is true.
Replica discover max hops	Maximum number of instances that should be contacted while attempting to find the backup of a session. The range of value is 1 to 2147.0483647.0, or -1 for no limit.
Startup discover timeout	Maximum time (in seconds) that an instance will spend trying to contact its designated backup instance. The range of value is 0.001 to 3600.
Cookie name	Enter the name of the cookie that tracks which instance owns a session.

Managing Authentication Realms

The Java EE based security model provides security realms that identify and authenticate users.

The authentication process verifies users through a Java realm. 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 decisions.

Note – The Java realms are analogous to the auth-dbs (Authentication Databases) with the difference that while auth-dbs are used by the ACL engine (based on rules in your ACL file), the Java Realms are used by the Java Servlet access control rules that are specified in each web application's web.xml file.

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

The following table defines the different types of realms supported in Web Server 7.0.

TABLE 11–8 Types of Realms

Realm	Description
File	The `file` realm is the default realm when you first install the Sun Java System 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 Java Realms are analogous to the auth-dbs (Authentication Databases) with the difference that while auth-dbs are used by the ACL engine (based on rules in your ACL file), the Java Realms are used by the Java Servlet access control rules that are specificed in each web application's web.xml.
LDAP	The `ldap` realm enables 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. You can also perform this action from your LDAP directory product's user management console.
PAM	The PAM (aka Solaris) realm delegates authentication to the Solaris PAM stack. As with the PAM auth-db, this realm is only supported on Solaris 9 and 10 and the server instance must be running as root.

TABLE 11–8 Types of Realms *(Continued)*

Realm	Description
Certificate	The `certificate` realm supports SSL authentication. The certificate realm sets up the user identity in the Sun Java System Web Server's security context and populates it with user data from the client certificate. The Java EE 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.
Native	The `native` realm is a special realm that provides a bridge between the core ACL-based authentication model and the Java EE/Servlet authentication model. By using the `native` realm for Java web applications it is possible to have the ACL subsystem perform the authentication instead of having the Java web container do so, thus leaving the `native` realm 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.
Custom	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.

The following section describes the steps involved in adding a new authentication realm.

▼ To Add a Authentication Realm

1. **Click the Configurations tab and select the configuration from the list.**

   Select the configuration for which you need to add a new authentication realm. and select the configuration.

2. **Click Java > Security tab.**

3. **Click New Authentication .**

4. **Provide Realm Details.**

   - **Name** — Enter a short name for the realm. This name is used to refer to the realm from, for example, web.xml.

- **Class** — If you are configuring a custom realm, enter the full Java class name which implements your custom realm. There is no need to enter a class for any of the built-in realms.
- **Type** — Select the type of realm. See the previous section where Java Realm types are discussed.
- **Properties** — Add realm specific properties. For instance, `property name="file" value="instance_dir/config/keyfile" and property name="jaas-context" value="fileRealm`.

Note – Using CLI

To add an authentication realm through CLI, execute the following command.

```
wadm> create-auth-realm --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 basic
```

See CLI Reference, `create-auth-realm(1)`.

Specify the name of a built-in authentication realm type. The type can be `file`, `ldap`, `pam`, `native` or `certificate`.

◆ ◆ ◆ CHAPTER 12

Working With Search Collections

The server includes a search feature that enables 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.

For more information on querying the search collections, refer to the *search online help*.

About Search

The search feature is installed with other web components during the installation of Sun Java System Web Server. Search is configured and managed at the virtual server level instead of the server instance level

From the Search tab under the Virtual Servers tab in the administration console, you can:

- Enable and disable the search feature
- Create, modify, delete, and re-index 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. Customization can include re-branding the pages with a corporate logo, or changing the way search results appear. In previous releases customization was accomplished through the use of pattern files.

There is no global "on" or "off" functionality for search. 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.

Note – The Sun Java System Web Server 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.

Configuring Search Properties

Search is enabled for a virtual server by enabling the search application included on the server.

Note – The Java web container must be enabled for search to be enabled.

After ensuring that Java is enabled for the virtual server you want to configure, enable search by performing the following steps:

1. Click the **Configurations** tab.
2. Select the configuration from the configuration list.
3. Click the **Virtual Servers** tab.
4. Select the virtual server from the virtual server list.
5. Click the **Search** tab.
6. Under **Search Application**, select the **Enabled** checkbox to enable the search application.

Other parameters which you can configure are listed below:

- **URI**. If you plan to use a custom search application, enter the URI; if you are using the default search application, you do not need to specify a value here.
- **Max Hits**. Specify the maximum results retrieved in a search query.
- **Enabled**. Select this to enable the default search application.

Note – Using CLI

To set search properties through CLI, perform the following command in CLI:

```
wadm> set-search-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8888 --no-ssl --rcfile=null --config=config1 --vs=config1_vs_1
enabled=true max-hits=1200
```

See CLI Reference, `set-search-prop(1)`.

Configuring 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.

Note – About Search 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
- Documents in a search collection are not specific to any one character encoding, which means that a search collection can be associated with multiple encoding.

Supported Formats

Files of the following format can be indexed and searched.

1. HTML documents — .html and .htm
2. ASCII Plain Text — .txt
3. PDF.

Adding a Search Collection

To add a new collection, perform the following tasks:

1. Click the **Configurations** tab.
2. Select the configuration from the configuration list.

3. Click the **Virtual Servers** tab.
4. Select the virtual server from the virtual server list.
5. Click the **Search** tab.
6. Under **Search Collections**, click **Add Search Collection** to add a new search collection.

The following section describes the fields in the page for creating a new search collection:

1. **Provide Search Collection Information**
   a. **Collection Name** — Enter a unique name for the search collection.

   Note – Multi byte characters are not allowed as collection name.

   b. **Display Name** — (Optional) The display name will appear as the collection name in the search query page. If you do not specify a display name, the collection name serves as the display name.
   c. **Description** — (Optional) Enter text that describes the new collection.
   d. **Path** — You can either create the collection in the default location or provide a valid path, where the collection will be stored.
2. **Provide Indexing Information**
   a. **Directory to Index** — Enter the directory from which documents will be indexed into the collection. Only the directories visible from this virtual server can be indexed.
   b. **Sub Directory**— Enter the sub directory from which documents will be indexed into the collection. Sub directory path should be relative to the directory path specified earlier.
   c. **Pattern** — Specify a wildcard to select the files to be indexed.

      Use the wildcard pattern judiciously to ensure that only specific files are indexed. For example, specifying *.* might cause even executable and Perl scripts to be indexed.
   d. **Subdirectories**— Enabled/Disabled. Default value is Enabled. If you enable this option, documents within the subdirectories of the selected directory will also be indexed.
   e. **Default Document Encoding** —

      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.
3. **Step 3: View the Summary**
   a. View the summary and click **Finish** to add the new collection.

Note – Using CLI

To add a search collection through CLI, execute the following command.

```
wadm> create-search-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/search_config1
--document-root=../docs searchcoll
```

See CLI Reference, `create-search-collection(1)`.

Deleting a Search Collection

To delete a search collection, perform the following tasks:

1. Click the **Configurations** tab.
2. Select the configuration from the configuration list.
3. Click the **Virtual Servers** tab.
4. Select the virtual server from the virtual server list.
5. Click the **Search** tab.
6. Under **Search Collections**, select the collection name and click **Delete** to delete the collection.

Note – Using CLI

To delete a search collection through CLI, execute the following command.

```
wadm> delete-search-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 searchcoll
```

See CLI Reference, `delete-search-collection(1)`.

Scheduling Collection Update

You can schedule maintenance tasks to be performed on collections at regular intervals. The tasks that can be scheduled are re-indexing and updating. The administrative interface is used to schedule the tasks for a specific collection. You can specify the:

- Task to perform (re-indexing or updating)
- Time of day to perform the task
- Day(s) of the week to perform the task

To schedule events for the collection, perform the following tasks:

1. Click the **Configurations** tab.
2. Select the configuration from the configuration list.
3. Click the **Virtual Servers** tab.
4. Select the virtual server from the virtual server list.
5. Click the **Search** tab.
6. Click the **Scheduled Events** tab.
7. Under **Search Events** tab, click **New**.

The following table describes the fields in the New Search Event Schedule page:

TABLE 12–1 Field Description > New Search Event Schedule

Field	Description
Collection	**Select the collection from the drop-down list for which you need to schedule maintenance.** 1. Re-index Collection—This scheduled event will re-index the specified collection at the specified time. 2. Update 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. This scheduled event will update the collection at the specified time. 3. Pattern—Specify a wildcard to select the files to be indexed. 4. Subdirectories included—If you select this option, documents within the subdirectories of the selected directory will also be indexed. This is the default action. 5. 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.**
Event	**The configured time when the event will start. Select the hour and minutes value from the drop-down box.** **Every Day — Starts the event specified every day at the specified time.** **Specific Days — Starts the event specified at specific days.** 1. *Days* — **Specify any day from Sunday to Saturday.** 2. *Dates* — **Specify any day of the month from 1 to 31 as comma separated entries. E.g. 4,23,9** **Specific Months — Starts the event specified at the specific time and month. Specify month from January to December.**
Time	**Start the specified event after this time period.** 1. *Every Hours* — **Select the number of hours from the drop-down box.** 2. *Every Seconds* — **Select the number of seconds from the drop-down box.**
Interval	

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 the Sun Java System 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.

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.

Note – For more detailed information about conducting basic and advanced searches, including information about keywords and optional query operators, see the online Help provided with the search engine. To access this information, click the Help link on the Search page.

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:

▼ Making a Query

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, select 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 Enter, or click Search) 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.

To make an advanced search query, perform the following steps:

▼ To Make an Advanced Search Query

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 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.

Document Field

The Sun Java System Web Server maintains an index of documents. The index contains an entry for each document. Each index entry contains one or more fields such as Title, Author, and URL. Queries can be limited to specific document fields, and documents are only found if they match your criteria in the specified fields.

For example, if you simply search for Einstein, you will find all documents that have the word Einstein in any one of the Title, Author, or Keywords fields. This will include documents about Einstein, documents that make reference to Einstein, and documents written by Einstein. But if you specify `Author = "Albert Einstein"` , you will only find documents written by Albert Einstein.

By default, the index fields that you can search are:

1. **Author** — The author, authors, or organization that created the document as specified with an `<author>` meta tag.
2. **Keywords** — The keywords as specified with a `<keywords>` meta tag.
3. **Date** — The date that this document was last edited or modified.
4. **Title** — The document's title as specified with the HTML `<title>` tag.

PDF files contain FTS information about the author, title and subject. To search in PDF files for these information, you can construct a query like `<title> contains Java`, *<subject> contains web server*.

Search Query Operators

For a detailed list of search query operators, refer to the *Administration Console Search Online Help*.

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. The matched words are also highlighted.

Customizing Search Pages

The Sun Java System 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 re-branding the web pages with a different logo, or as complex as changing the order in which search results are displayed.

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 `[install_dir]/lib/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.

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 check boxes 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. 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 check boxes, 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 creates 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 titled "Search", which uses the same format as other items on the sidebar.

In the sample code given below, the form body contains three check boxes 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: 50p
        </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=\\"Prev
         <s1ws:resultStat formId="test" type="range" />
         <s1ws:resultNav formId="test" type="next" caption="<img border=0 src=\\"images/arrow-right.gif\\" alt=\\"N
          <!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 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`. An example of the attribute for search results, 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

The Sun Java System 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 Java System Web Server 7.0 *Developer's Guide to Web Applications.*

◆ ◆ ◆ CHAPTER 13

Monitoring Your Server

This section describes the monitoring capabilities of the Sun Java System Web Server and provides a detailed list of the server parameters you can monitor at both instance and configuration level.

Monitoring Capabilities in Sun Java System Web Server

The server parameters that can be monitored are displayed when you select the Configurations or Instances tab under the monitoring parent tab.

From the Sun Java System Web Server Administration Console, you can perform the following actions:

- View server statistics at an instance and configuration level.
- Enable/Disable monitoring at configuration level.
- View error/access log at the instance level.

To monitor server parameters at the configuration level, click Monitoring > Configurations tab. The table lists the available configuration along with the following information:

- **Nodes** — The number of nodes in which the configuration has been deployed.
- **Requests** — Total number of requests received across all virtual servers.
- **Errors** — Total number of errors logged across all virtual servers.
- **Response Time** — The maximum response time for any of the virtual servers.

Click the configuration name to see the configuration level statistics. The general statistics are divided into three types:

- Request Statistics
- Error Statistics
- Response Time Statistics

Monitoring The Server Statistics

The server statistics can be viewed across the following categories:

- General Statistics
- Instance Statistics
- Virtual Server Statistics

TABLE 13–1 Monitoring Categories

Category	Description
General Statistics	General Statistics shows overall Request, Error and Response statistics for the configuration.
Instance Statistics	Instance Statistics shows overall Request, Error and Response statistics for the instances along with information on server crash and virtual server count.
Virtual Server Statistics	Virtual Server Statistics shows overall Request, Error and Response statistics for the virtual servers along with the number of open connections and total bytes received/transmitted.

▼ Viewing The Statistics

1. **Click the Monitoring tab.**
2. **Select the configuration from the list.**
3. **View General, Instance and Virtual Server Statistics.**

Note – Using CLI

You can monitor the server using the `get-config-stats`, `get-virtual-serevr-stats`, `get-webapp-stats` and `get-servlet-stats` commands.

- `wadm> get-config-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --node=cat.test.com --ssl=true`

  The preceding command will fetch the statistics for the given instance. To see the statistics at the configuration level, the above command can be used without the `--node` option.

- `wadm> get-virtual-server-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --vs=www.test.com --node=cat.test.com --ssl=true`

  The preceding command will fetch the aggregated virtual server statistics for a given configuration across all the nodes where the configuration has been deployed. To see the statistics for a configuration deployed on a given node `--node` option can be used.

- `wadm> get-webapp-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --node=cat.test.com --vs=www.test.com --uri=/foo --ssl=true`

  The preceding command will fetch the statistics for a given web application deployed on the given virtual server of the given instance. To see the aggregated web application statistics for a given configuration across all the nodes where the configuration has been deployed, the previous command can be used without the `--node` option.

- `wadm> get-servlet-stats --user=admin --password-file=admin.pwd --host=localhost --port=8989 --config=test --node=cat.test.com --vs=www.test.com --uri=/servlet-simple --ssl=true`

  The preceding command will fetch the statistics for the servlet `servlet-simple`.

▼ Viewing the Monitoring `stats-xml` File

1. **From the Common Tasks page, click the Configuration tab and select the configuration from the list.**
2. **Click the Edit Virtual Server tab.**
3. **Click the Monitoring Setting tab.**
4. **Enable the XML Report check box and provide the publishing URI.**
5. **Click the Save button.**

6 **Click the Deployment Pending link at the top right of the screen.**

7 **Click the Deploy button.**

For example, if you have configured the default URI, then you can view the stats-xml file by typing the following URL in the browser.

http://*host:port*/`stats-xml`

Note – If you want to view the `.dtd` of the stats-xml file, type the following URL in the browser.

http://*host:port*/`stats-xml/yyy.dtd`

Modifying Monitoring Parameters

The server performs monitoring actions through SNMP. SNMP is a protocol used to exchange data about network activity. With SNMP, data travels between a managed device and a network management station (NMS). 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 the Sun Java System Web 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 Java System Web Server except for the Administration Server has a subagent.

Note – After making any SNMP configuration changes, you must click the Save button, then restart SNMP subagent.

To change settings for the configuration, perform the following tasks:

1. Click the **Configurations** tab.
2. Select the configuration for which you need to change monitoring settings.
3. Click the **Monitoring Settings** sub tab.

Configuring Monitoring Parameters

To change general monitoring settings for a configuration, edit the values under the General Settings section. The following table provides the field description of general monitoring parameters:

TABLE 13–2 Field Description > General Monitoring Settings

Field	Description
SNMP subagent	To use SNMP you must have a master agent and at least one subagent installed and running on your system. You need to install the master agent before you can enable a subagent. Select this option to enable/disable SNMP subagent.
Interval	The poll interval is the number of seconds between updates of the statistics information displayed. If your server instance is running, and you have enabled statistics, 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.
Profiling	You can use the statistics/profiling 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. Select this option to enable/disable profiling.

Configuring SNMP Subagent Parameters

To change SNMP subagent settings for a configuration, edit the values under the SNMP Subagent Settings section. The following table provides the field description of SNMP Subagent parameters:

TABLE 13–3 Field Description > SNMP Subagent Settings

Field	Description
Enabled	To use SNMP you must have a master agent and at least one subagent installed and running on your system. You need to install the master agent before you can enable a subagent. Select this option to enable/disable SNMP statistics collection.
Master Host	Enter the name and domain of the server (*UNIX only*).

TABLE 13–3 Field Description > SNMP Subagent Settings *(Continued)*

Field	Description
Description	Enter a short description for the server including operating system information.
Organization	Enter a short name representing the organization.
Location	Enter the location information of the server in this field.
Contact	Enter the contact information of the server in this field.

Configuring SNMP Subagent

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 Sun Management Center 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 Java System Web 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.

To start the SNMP subagent, perform the following tasks:

1. Click the **Nodes** tab.
2. Select an available node from the nodes list.
3. Click the **SNMP Subagent** tab.
4. Click **Start SNMP Subagent** to start the subagent.

Note – Before starting the SNMP subagent, verify that the master agent is running. The subagent is started only when the master agent is running.

To stop the SNMP subagent, perform the following tasks:

1. Click the **Nodes** tab.
2. Select an available node from the nodes list.

3. Click the **SNMP Subagent** tab.
4. Click **Stop SNMP Subagent** to stop the subagent.

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. The following table 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:

- that your system is already running an SNMP agent (an agent native to your operating system)
- that your native SNMP agent supports SMUX communication (If you're using the AIX platform, your system supports SMUX.)

See the 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 13–4 General Guidelines

If your server meets these conditions	Follow these procedures
■ 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.

TABLE 13–4 General Guidelines *(Continued)*

If your server meets these conditions	Follow these procedures
■ Native agent is currently running ■ SMUX supported	1. Reconfigure the SNMP native agent. 2. Enable the subagent for each server installed on the system.

Configuring SNMP Using CLI

▼ To Activate SNMP on Solaris

1 Configure SNMP Parameters.

Set the SNMP parameters for the configuration.

```
wadm> enable-snmp --user=admin --password-file=../admin.passwd
--host=serverhost --port=8989 --ssl=true --no-prompt --rcfile=null
--config=config1 --loconfig1ion=india --master-host=hostname
--description=cli-snmp --organization=sun --contact=internal
```

2 Deploy the Configuration.

```
wadm> deploy-config --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 config1
```

3 Start the Server Instance.

```
$ ./https-test/bin/startserv
```

4 Run the Master Agent (magt) as root.

Note – To run `magt`, native snmpd must be stopped.

```
$ cd /etc/init.d/
          $ init.dmi stop; init.snmpdx stop; init.sma stop
```

Remove the file `https-admserv/config/logs/pid.masteragt` (If present).

```
$ rm ./https-admserv/config/logs/pid.masteragt
          wadm>  start-snmp-master-agent --snmp-port 161 hostname
```

5 Start the Sub Agent.

Remove the file `https-admserv/config/logs/pid.httpagt`(If present).

```
$ rm ./https-admserv/config/logs/pid.httpagt
```

Kill the `httpagt` if it is already running

```
wadm> start-snmp-subagent hostname
```

To Activate SNMP on Linux

1 Configure SNMP Parameters.

Set the SNMP parameters for the configuration.

```
wadm> enable-snmp --user=admin --password-file=../admin.passwd
--host=serverhost --port=8989 --ssl=true --no-prompt --rcfile=null
--config=config1 --loconfig1ion=india --master-host=hostname
--description=cli-snmp --organization=sun --contact=internal
```

2 Deploy the Configuration.

```
wadm deploy-config --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 config1
```

3 Start the Server Instance.

```
$ ./https-test/bin/startserv
```

4 Run the Native Master Agent (snmpd) as root.

To allow direct communication with snmpd , add the following line in /etc/snmp/snmpd.conf and restart snmpd.

```
smuxpeer 1.3.6.1.4.1.42.2.190.1
```

```
view systemview included .1.3.6.1.4.1.42.2.190.1
```

```
# cd /etc/init.d/
      # ./snmpd stop
      # ./snmpd start
```

5 Start the Sub Agent.

Remove the file `https-admserv/config/logs/pid.httpagt`(If present).

```
$ rm ./https-admserv/config/logs/pid.httpagt
```

Kill the `httpagt` if it is already running

```
wadm> start-snmp-subagent hostname
```

To Activate SNMP on Windows

1. **Configure SNMP Parameters.**

   Set the SNMP parameters for the configuration.

   ```
   wadm> enable-snmp --user=admin --password-file=../admin.passwd
   --host=serverhost --port=8989 --ssl=true --no-prompt --rcfile=null
   --config=config1 --loconfig1ion=india --master-host=hostname
   --description=cli-snmp --organization=sun --contact=internal
   ```

2. **Deploy the configuration.**

```
<install_root>\bin\wadm.bat deploy-config --user=admin --host=<hostname> --port=8989 <config1>
```

   You can check whether the SNMP service has been enabled, by running the command:

   ```
   <install_root>\bin>wadm.bat get-snmp-prop --user=admin --port=8989 -c <config1>

   contact=internal
   enabled=true
   description=snmp
   master-host=127.0.0.1
   location=us
   organization=sun
   ```

3. **Start the Web Server instance using Windows Services option.**

4. **Configure both SNMP and SNMP Trap Services according to the MSDN document.**

5. **Start SNMP Service and SNMP Trap Service using Windows Services option.**

Note – Ensure that `<install_root>/lib` directory is present in the System Path environment variable.

To Configure Peer Based Master Agent (magt)

You can configure peer based master agent to integrate with OS Native Master Agent on Solaris 10 and Linux by following these steps.

Note – The Solaris 10 OS Native Master Agent is snmpd. By default it runs on SNMP default UDP port 161. This agent is configurable using the /etc/sma/snmp/snmpd.conf file. The agent provides a proxy directive for forwarding the request/response to other Master Agents or to a Subagent. For more information, refer to the snmpd.conf manual page.

For Solaris 8 and 9, there is no clean integration with the OS Native Master Agent snmpd. For Linux, the httpagt can directly integrate with snmpd, in which case there is no need to run magt. For Windows, the Sun Java System Web Server snmp library directly communicates with windows SNMP service.

1. **Start the master agent specifying the SNMP port (11161) as mentioned in the note above.**

2. **Add the following in /etc/sma/snmp/snmpd.conf for Solaris 10 .**

   ```
   proxy -v 1 -c public myserver:11161 .1.3.6.1.4.1.42.2.190.1
   ```

3. **Restart the snmpd.**

   ```
   # cd /etc/init.d
   # init.dmi stop; init.snmpdx stop; init.sma stop
   # init.dmi start; init.snmpdx start; init.sma start
   ```

4. **To get the SNMP data use the snmpwalk on port:**

   ```
   $ snmpwalk -c public -v 1 <host-name>:<port> 1.3.6.1.4.1.42.2.190.1
   ```

Setting Up Logging for Your Server

The Administration Server log files record data about the server, including the types of errors encountered and information about server access. Viewing these logs enables 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 from the administration console. 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 Log 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.

Types of Log

The log type can be broadly classified as:

1. **Access Log** — The access log records information about requests to and responses from the server.
2. **Server Log** — The Server 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.

Viewing Access and Server Logs

- **Viewing Server Logs**.

  ```
  wadm> get-log --user=admin --password-file=admin.passwd --host=localhost
  --port=8989 --start-date=01/01/2006:09:00:00 --end-date=04/01/2006:10:00:00
  --config=test cat.test.com
  ```

  The preceding command displays the Server Logs of a given configuration between the date `01/Jan/2006:09:00:00` and `04/Jan/2006:10:00:00`.

- **Viewing Access Logs**.

  ```
  wadm> get-access-log --user=admin --password-file=admin.passwd
  --host=localhost --port=8989 --status-code=300 --config=test cat.test.com
  ```

  The preceding command will display only those access log entries of a given configuration which have a status code of 300.

In the preceding commands, the start-date and the end-date options must be in the following format — `dd/MM/yyyy:HH:mm:ss`. The date format can also be customized. You can use a variable `wadm_log_date_format` in the rcfile to specify your own date format rather than using the default date format.

In the Sun Java System Web Server, you can enable access log by executing the following command:

```
enable-access-log --user=admin --host=serverhost
--password-file=../admin.passwd --port=8989 --ssl=true --no-prompt
--rcfile=null --config=config1 --vs=vs --uri-pattern="*.html"
--file=../logs/access.new --log-ip=true--format="%Req->reqpb. protocol%
%Req->headers.authorization% %vsid% %Ses->client.dns%"
```

Configuring Log Parameters

To enable and edit log settings for a configuration, perform the following tasks:

1. Click the **Configuration** tab.
2. Select the configuration for which you will need to enable/edit log settings.
3. Click the **General Settings** > **Log Settings** tab.

Editing Access Log Preferences

The fields in the Access Log Preferences section are described in the following table:

TABLE 13–5 Field Description > Editing Access Log Preferences

Field	Description
Access Log	*Enabled/Disabled*. By default, access log is enabled. Select this option to disable access log. Note that enabling access log will degrade server performance at a very low magnitude.
File Location	The server path, where the access log files will be stored. Default values is *../logs/access*
Log Format	1. Use Common Log Format — This option is the default format type for the log file. The server will log most relevant information extracted from the request headers. Common log format is `IP address – user [date] "request" status content-length`. 2. Log these details only — You can log only specific values from the request header using this option. Select one of the following values: ■ Client Hostname ■ System Date ■ HTTP Status ■ HTTP Header ■ HTTP Method ■ Query String ■ Virtual Server Name ■ Authenticated User Name ■ Complete HTTP Request ■ Content Length ■ Request URI ■ Protocol

Editing Server Log Preferences

The fields in the Server Log Preferences section are described in the following table:

TABLE 13–6 Field Description > Editing Server Log Preferences

Field	Description
Server Log Location	The server path, where the Server Log files will be stored. The default value is *../logs/errors*

TABLE 13–6 Field Description > Editing Server Log Preferences *(Continued)*

Log Verbosity Level	This option provides a means of setting log granularity. To test and debug web applications, the recommended level is *finest*. For a production environment, the recommended log level is *failure* or *security*. *catastrophe* log level will log very few details.
Log Virtual Server Name	If this option is selected, the name of the virtual server processing the request is logged along with any errors.
Log to System Log	Logs all messages to the system log.
Log to console	If this option is selected, exceptions arising from deployed web applications are logged, if they are written to *console*. This option is enabled by default.
Date Format	The time format, which is used to append time stamps to the error messages. The default value is *[%d/%b/%Y:%H:%M:%S]*

Archiving Log Files

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

For example, you can set up files to rotate every hour, and the server saves and names the file "`access.199907.0152400`," 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, the server creates a time-stamped access log file and rotation starts at server startup.

Once the rotation starts, the 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."

Setting Log Rotation

You can create a schedule for error/access log rotation for the configured instances by using the log rotation option. To set up log rotation, perform the following steps:

1. Click the **Configuration** tab.
2. Select the configuration for which you need to enable/edit log settings.
3. Click the **General Settings** > **Log Settings** tab.
4. Under **Log Archiving**, click **New**.

The fields in the new log rotation page is described in the following section:

TABLE 13–7 Field Description > Setting Log Rotation

Field	Description
Event	*Access Log Rotation / Server Log Rotation*. Select any or both of these options to configure rotation for that log type.
Time	The configured time when the event will start. Select the hour and minutes value from the drop-down box.
	Every Day — Starts the event specified every day at the specified time.
	Specific Days — Starts the event specified at specific days.
	1. *Days* — Specify any day from Sunday to Saturday.
	2. *Dates* — Specify any day of the month from 1 to 31 as comma separated entries. E.g. 4,23,9
	Specific Months — Starts the event specified at the specific time and month. Specify month from January to December.
Interval	Start the specified event after this time period.
	1. *Every Hours* — Select the number of hours from the drop-down box.
	2. *Every Seconds* — Select the number of seconds from the drop-down box.

If you need to delete the scheduled log rotation, Click **Delete** in the **Log Archiving** section.

Archive Command

You can specify the absolute path of the command after the server rotates the log file. The post-rotation filename of the log file is passed as an argument to the archive command. The archive command also compresses the log file that has been rotated.

Configuring Log Settings for Administration Server

All the configuration changes performed using the administration console are logged by the administration server. Some common actions logged are creating new configurations, creating virtual servers, and configuring instance settings. However configuration level details like accessing a web application or exceptions raised while accessing a web application are logged separately by the configuration.

▼ To Modify the Server Log Location

1. **Click the Administration Server > General tab.**
2. **Go to the Log Preferences section.**
3. **Edit the Server Log Location field.**

   Log the location where the errors will be stored. Provide a valid server path for maintaining the log files. Also check if the administration server has the permission to write in the specified directory for UNIX systems.

   The default location is `../log/errors`

▼ To Modify the Log Verbosity Level

1. **Click the Administration Server > General tab.**
2. **Go to the Log Preferences section.**
3. **Select the Log Verbosity Level.**

   This option provides you with a means of setting log granularity. For testing and debugging, the recommended level is *finest*.

   For a production environment, the recommended log level is *failure* or *security*. *catastrophe* log level will log very few details.

▼ To Modify the Date Format for the Log

1. **Click the Administration Server > General tab.**
2. **Go to the Log Preferences section.**
3. **Edit the Date Format Field.**

   The time format, which will be used to append time stamps to the error messages. The default value is *[%d/%b/%Y:%H:%M:%S]*

◆ ◆ ◆ CHAPTER 14

Internationalization and Localization

The internationalized and localized version of the Sun Java System Web Server provides support for multiple languages and multiple encodings.

Entering Multi-byte Data

If you want to enter multi-byte data on the administration console 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 multi-byte characters.

LDAP Users and Groups

For email addresses, use only those characters permitted in RFC 17.000 (`ftp://ds.internic.net/rfc/rfc17.000.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 to input 8 bit or multi-byte data.

Support for Multiple Character Encodings

The Sun Java System Web Server 7.0 provides multiple character encoding support for the following features:

WebDAV

The Sun Java System Web Server supports setting and retrieving multi-byte properties in the `PROPPPATCH` and `PROPFIND` methods. While requests can be made in any encoding format, the response from the server is always in UTF-8.

Search

The Sun Java System Web Server 7.0 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 meta tags 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 Java System Web Server 7.0 *Developer's Guide to Web Applications*.

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 enabling the **Negotiate Client Language** checkbox under **Configuration** > **(Select Configuration)** > **Virtual Server** >**(Select Virtual Server)** > **Server Setting** > **General** > **Localization**.

For example, if this option is enabled, 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
```

then your server searches for the file in the following order:

Search 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 – When naming localized files, keep in mind 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 according to the algorithm illustrated above.

◆ ◆ ◆ APPENDIX A

CLI Changes From Previous Version

The following table depicts some common tasks that can be performed using the Sun Java System Web Server 7.0 and the earlier version.

TABLE A–1 CLI changes from previous version

Task	6.1 CLI	7.0 Update 2 CLI
List all the deployed web applications for an instance.	`wdeploy list -i INSTANCE_NAME -v VIRTUAL_SERVER`	`wadm> list-webapps --user=admin --port=8888 --password-file=admin.passwd --no-ssl`
Deploy a anew web application.	`wdeploy deploy -i INSTANCE_NAME -v VIRTUAL_SERVER -u URI_PATH war file name`	1. `wadm> add-webapp --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs=VIRTUAL_SERVER --uri=URI_PATH war file name` 2. `wadm> deploy-config --user=admin --port=8888 --password-file=admin.passwd 'HOSTNAME'`
Reconfiguring a running instance.	No support.	`wadm> reconfig-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME`
List all virtual servers for an instance.	`HttpServerAdmin list -v -d INSTALL_DIR -sinst https-INSTANCE_NAME`	`wadm> list-virtual-servers --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME`

TABLE A–1 CLI changes from previous version *(Continued)*

Task	6.1 CLI	7.0 Update 2 CLI
List all JDBC resources.	`HttpServerAdmin list -r -jdbc -d INSTALL_DIR -sintance https-INSTANCE_NAME`	`wadm> list-jdbc-resources --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME`
Create a custom resource.	`HttpServerAdmin create -r -custom -jndiname -poolname -enabled true`	`wadm> create-custom-resource --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --res-type=type --jndi-name NAME`
Start an instance.	No support.	`wadm> start-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*`
Stop an instance.	No support.	`wadm> stop-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*`
Configuring web server with reverse proxy.	No support.	1. `wadm> create-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --from='URI' --server='target-hostname'` 2. `wadm> set-reverse-proxy-prop --user=admin --password-file=admin.pwd --host=serverhost --port=8888 --config=config1 --vs=config1_vs_1 --uri-prefix=/test/ --server=http://java.com:8080 --sticky-cookie=testCookie`
Disable reverse proxy.	No support.	`wadm> delete-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --uri-prefix='URI'`

TABLE A–1 CLI changes from previous version *(Continued)*

Task	6.1 CLI	7.0 Update 2 CLI
Enable WebDAV.	No support.	1. `wadm> enable-webdav --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME` 2. `wadm> deploy-config --user=admin --port=8888 --password-file=admin.passwd HOSTNAME`
Create a new web server	No support.	1. `wadm> create-config --doc-root=[DOCROOT] --jdk-home=[JAVAHOME] --server-user=[SERVERUSER] [--document-root=serverdocroot] [--platform=32|64] --http-port=port --server-name=servername CONFIGNAME` 2. `wadm> create-instance --config=CONFIGNAME NODENAME` 3. `wadm> deploy-config CONFIGNAME`

◆ ◆ ◆ APPENDIX B

FastCGI Plug-in

Introduction

FastCGI is an enhancement to the existing CGI (Common Gateway Interface), which is a standard for interfacing external applications with Web Servers. Like CGI, FastCGI applications run in separate, isolated processes. Some of the advantages of using FastCGI are as follows:

- Enables applications to persist between client requests, eliminates application start up overhead, and enables the application to maintain state between client calls.
- Enables applications to reside on remote systems (a different system from where the Web Server is running).
- Enables additional flexibility in application functionality, with explicit support for applications that do client authentication and filtering of input.
- enables the administrator to restrict the impact on the system that is caused by the FastCGI servers.

FastCGI plug-in enables Web Server to safely work with popular third-party dynamic content generation technologies (such as Perl and Python) in a scalable way.

For more information on FastCGI, refer to the specification at http://www.fastcgi.com/devkit/doc/fcgi-spec.html.

Plug-in Functions (SAFs)

FastCGI plug-in provides the following Server Application Functions (SAFs):

The various parameters and "error-reason" strings for the FastCGI SAFs are described in the following sections:

auth-fastcgi

`auth-fastcgi` is a PatchCheck function. This function is used to forward the request to an "Authorizer" FastCGI application. On successful authorization, a return code of 200 is sent. Otherwise, the response from the "Authorizer" FastCGI application is sent back to the user agent.

More information on the FastCGI Roles can be found here http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6.

The parameters accepted by `auth-fastcgi` SAF, are available at: "FastCGI SAF Parameters" on page 245.

The following `obj.conf` code example demonstrates the use of `auth-fastcgi`:

```
PathCheck fn="auth-fastcgi" app-path="/usr/bin/perl"
app-args="/fastcgi/apps/auth/SimpleAuth.pl" bind-path="localhost:3432".
```

responder-fastcgi

The `responder-fastcgi` is a Service function. This function is used to forward the request to a FastCGI application that acts as a "Responder". The response from the Responder application is sent back to the user agent. More information on the FastCGI Roles can be found at http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6.

The list of parameters accepted by `responder-fastcgi` SAF are available at:"FastCGI SAF Parameters" on page 245 .

The following `obj.conf` code example demonstrates the use of `responder-fastcgi`:

```
Service fn="responder-fastcgi"
app-path="/fastcgi-enabled-php-installation/bin/php" bind-path="localhost:3433"
app-env="PHP_FCGI_CHILDREN=8" app-env="PHP_FCGI_MAX_REQUEST=500".
```

filter-fastcgi

The `filter-fastcgi` is a Service function. This function is used to forward the request to a "Filter" type of FastCGI application. The "Filter" application receives the information associated with the HTTP request and also the data from the file stored on the server. The "Filter" application then generates a "filtered" version of the data stream as the response. This response is sent back to the user agent. More information on the FastCGI Roles can be found at http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6.

The list of parameters accepted by `filter-fastcgi` SAF are available at:"FastCGI SAF Parameters" on page 245 .

The following `obj.conf` code example demonstrates the use of `filter-fastcgi`:

```
Service fn="filter-fastcgi" app-path="/fastcgi/apps/filter/SimpleFilter"
bind-path="localhost:3434"
app-env="LD_LIBRARY_PATH=/fastcgi/fcgi-2.4/libfcgi/.libs" min-procs=2
```

error-fastcgi

The `error-fastcgi` is an Error function. The `error-fastcgi` SAF handles the errors specific to the FastCGI plug-in. This function however does not handle the HTTP errors. On error, FastCGI plug-in can be configured to display a specific page or redirect the request to a specific URL.

The list of parameters accepted by `error-fastcgi` SAF, are available at: "FastCGI SAF Parameters" on page 245.

The following `obj.conf` snippet demonstrates the use of `error-fastcgi`:

```
Error fn="error-fastcgi" error-reason="Invalid Parameters"
error-url="http://www.foo.com/errorPage.html"
```

See "FastCGI SAF Parameters" on page 245 for information on the `error-fastcgi` parameters.

FastCGI SAF Parameters

The FastCGI plug-in SAFs, "`auth-fastcgi`" , "`responder-fastcgi`" and "`filter-fastcgi`", all accept the following parameters unless otherwise mentioned explicitly:

- `bind-path` - (Optional) Can be a UNIX Domain Socket name or Named Pipes or of the form `host:port`. The description of `app-path` parameter explains the usage of `bind-path` parameter.
- `app-path` - (Optional) FastCGI application path that processes the request. The functionality is dependent on the value of the `bind-path` parameter as follows:

1. If only `app-path` is specified, the plug-in creates FastCGI applications that listen to UNIX Domain Sockets or Named Pipes created by the plug-in.
2. If both `app-path` and `bind-path` are specified, the plug-in starts the specified FastCGI application process and binds them to the specified `bind-path`.
3. If only `bind-path` is specified, the FastCGI application is considered to be running remotely and the plug-in will not start the FastCGI application process.
4. If "`app-path`" and "`bind-path`" both are not specified, then the plug-in logs an error message.

- `app-args` — (Optional) Values that are passed as arguments to the FastCGI application process. Multiple `app-args` parameters are allowed. The format for the multiple `app-args` parameters is `app-args="value" app-args="value" ...`
- `app-env` - (Optional) Value pairs that are passed as environment variables to the FastCGI application process. Multiple "`app-env`" parameters are allowed. The format for multiple `app-env` parameters is `app-env="name=value" app-env="name=value"`. The existing Web Server environment variables are not passed on to the FastCGI programs. Hence, you should explicitly set the environment variables for FastCGI programs using `app-env`.

  To compile a PHP program, you should ensure that the library files are configured correctly.

  For example, if you want to load PHP binaries that you have compiled as a FastCGI application, you need to ensure that all the dependent library files, `/usr/local/lib` and `/usr/local/mysql/lib`, are exported to `LD_LIBRARY_PATH`.

  `app-env="LD_LIBRARY_PATH=/usr/local/lib:/usr/local/mysql/lib"`

  On Windows, `app-env="Path=c:/php/lib:c:/mysql/lib"`

  `app-env` also enables you to export other environment variables to the PHP applications. You can specify the `php.ini` file location as the following`app-env="PHPRC=<directory path>"`.

  While using PHP, you need to provide higher value to `PHP_FCGI_CHILDREN` and `PHP_FCGI_MAX_REQUESTS` so that it takes the higher precedence while configuring PHP with FastCGI.
- `min-procs` - (Optional) Integer specifying the minimum number of FastCGI application processes to be created. Defaults to 1.
- `max-procs` - (Optional) Integer specifying the maximum number of FastCGI application processes that can be created at any time. The integer value must be equal to or greater than `min-procs`. Defaults to 1.

Note – The default value is at present a non-operational parameter. For more information about this issue, see"FastCGI " in *Sun Java System Web Server 7.0 Update 4 Release Notes*.

- `reuse-connection` - (Optional) Boolean value that determines if connections to FastCGI applications are reused. `False` (`0`, `false`, `no`) indicates that the connections to FastCGI applications are closed after each request. `True` (`1`, `true`, `yes`) indicates that existing connections are reused for new requests. The default is `false`. See also `connection-timeout`.
- `connection-timeout` - (Optional) If "`reuse-connection` " is set to `True`, then this value specifies the timeout value in seconds for the pooled connections. If a connection is idle for the specified amount period of time, then the plug-in closes the connection. The default value for this parameter is 5 seconds. See also `reuse-connection` .
- `resp-timeout` - (Optional) Integer that represents the FastCGI server response timeout in seconds. If there is no response from the FastCGI application within the specified period of time, the request is discarded. The default value for this parameter is 5 minutes.
- `restart-interval` - (Optional) Integer that represents the time interval (in minutes) after which the FastCGI application is restarted. The default value for this parameter is 60 minutes (1 hour). If the value for this parameter is set to zero, the FastCGI application is not forced to restart.
- `req-retry` - (Optional) Integer that represents the number of times the plug-in should resend the request when the FastCGI application rejects the request. The default value for this parameter is zero.
- `listen-queue` - (Optional) Integer specifying the listen queue size for the socket. The default value for this parameter is 256.
- `rlimit_cpu` — Specifies the maximum amount of CPU time (in seconds) to be used by a FastCGI program. You can specify only the current (soft) limit. A maximum (hard) limit is not applicable for this parameter and will be ignored.

Note that parameters `chroot`, `user`, `group`, `nice`, `chdir`, `rlimit_as`, `rlimit_core` and `rlimit_nofile` are applicable to UNIX platforms only. On Windows platforms, these parameters are ignored.

- `chroot` - (Optional UNIX only) Used to set the root directory of the `chroot` FastCGI server application processes.
- `user` - (Optional UNIX only) Specifies the user ID the FastCGI application runs as. Defaults to Web Server's user ID.
- `group` - (Optional UNIX only) The FastCGI application will be running under the specified group. Defaults to Web Server's group.
- `nice` - (Optional UNIX only) Specifies the `nice`/ `priority` value of FastCGI application processes.
- `chdir` - (Optional UNIX only) Specifies the directory to `chdir` to after `chroot`, but before execution begins.

- `rlimit_as` - (Optional UNIX only) Specifies the maximum CGI program address space (in bytes). You can supply both current (soft) and maximum (hard) limits, separated by a comma. The soft limit must be listed first. If only one limit is specified, both the limits are set to this value.
- `rlimit_core` - (Optional UNIX only) Specifies the maximum CGI program core file size. A value of 0 disables writing cores. You can supply both current (soft) and maximum (hard) limits, separated by a comma. The soft limit must be listed first. If only one limit is specified, both the limits are set to this value.
- `rlimit_nofile` - (Optional UNIX only) Specifies the maximum number of file descriptors for the CGI program. You can supply both current (soft) and maximum (hard) limits, separated by a comma. The soft limit must be listed first. If only one limit is specified, both the limits are set to this value.

The `error-fastcgi` Server Application Function (SAF) accepts the following parameters:

- `error-url` - Specifies the page, URI or URL to be displayed in case of if a failure or error occurs. The value of this parameter can be an absolute path, a path relative to docroot, or an URL or URI.
- `error-reason` - (Optional) String that represents the FastCGI protocol error. This string is used to differentiate error URLs to be displayed, in case of any plug-in errors.

error-fastcgi SAF Error Reason Strings

This section provides a list of all the valid "error-reason" strings and their descriptions:

- “Missing or Invalid Config Parameters” : whenever app-path and bind-path are not specified.
- “Stub Start Error” : failure to start the Fastcgisub process.
- “Stub Connection Failure” : unable to connect to Fastcgistub.
- “No Permission” : FastCGI application or the Fastcgisub has no execute permission.
- “Stub Request Handling Error” : unable to send the request to stub, received invalid or no response from the stub for a request, and so on.
- “Set Parameter Failure” : when `set user, group, chroot, nice` or other parameters fail.
- “Invalid user and/or group” : when user or group is invalid.
- “Server Process Creation Failure” : FastCGI application execution failure or the FastCGI application is unable to bind to the specified address.
- “Fastcgi Protocol Error” : FastCGI application contains header with invalid FastCGI version or the role.
- “Internal Error” : unable to open the file to be sent to the filter application or any other unknown errors..

Configuring FastCGI Plug-in on Web Server

FastCGI plug-in is packaged with Web Server 7.0. You can configure FastCGI Plug-in on Web Server in one of following ways:

Configuring FastCGI Plug-in on Web Server Manually

The plug-in is installed at the following location:

32 bit FastCGI plug-in binaries are installed under <*install_dir*>`/plugins/fastcgi` directory. 64 bit Solaris SPARC FastCGI plug-in binaries are installed under <*install_dir*>`/lib/plugins/fastcgi/64` directory.

The following FastCGI binaries are installed :

`libfastcgi.so` (for Solaris/Linux)
`fastcgi.dll`(for Windows)
`Fastcgistub.exe`(for Windows)
`libfastcgi.sl`(for HP-UX)
`Fastcgistub` (executable)

The FastCGI plug-in is configured using the Web Server configuration files located under `<instance-dir>/config` directory. To configure the FastCGI plug-in, perform the following steps:

Modify the magnus.conf

Use the "load-modules" Init function to load the FastCGI plug-in shared library.

```
Init fn=flex-init access="access" format.access="%Ses->client.ip%
- %Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"

Init fn="load-modules" shlib="libJava EEplugin.so" shlib_flags="(global|now)"
```

```
Init fn="load-modules" shlib="libfastcgi.so" shlib_flags="(global|now)"
```

Modify the MIME Type (Optional)

Edit the `mime.types` file to specify the MIME mapping. Modifying the MIME type mapping is an optional step.

For Example,

```
#--Sun Microsystems Inc. MIME Information

# Do not delete the above line. It is used to identify the file type.

#

# Copyright 2006 Sun Microsystems, Inc. All rights reserved.

# Use is subject to license terms.

#


type=application/octet-stream exts=bin

type=application/astound exts=asd,asn

...

...

type=magnus-internal/fastcgi exts=php

...

...
```

Modify the obj.conf

Edit the obj.conf file to configure FastCGI specific requests using the plug-in SAFs described in the earlier sections.

An example of modified obj.conf file is shown below:

```
#

# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
```

```
# Use is subject to license terms.

#


# You can edit this file, but comments and formatting changes

# might be lost when you use the administration GUI or CLI.



<object name = "default">

        AuthTrans fn="match-browser" browser="*MSIE*"
                        ssl-unclean-shutdown="true"
        NameTrans fn="ntrans-Java EE" name="Java EE"
        NameTrans fn="pfx2dir" from="/mc-icons"
                        dir="/ws7/lib/icons" name="es-internal"
        NameTrans fn="assign-name" from="/fcgi/*" name="fcgi.config"

</object>

<Object name="fcgi.config">

        AuthTrans fn="auth-fastcgi" app-path="/fastcgi/apps/c/simpleAuth"
                bind-path="localhost:2111"
        Service fn="responder-fastcgi"
                        app-path="/fastcgi_enabled_php_installation_dir/bin/php"
                app-env="name1=abc"

</object>
...
```

Note that FastCGI SAFs can be invoked in different ways by defining different objects for different URL patterns or mapping the SAFs to different MIME types.

For more information on obj.conf configuration and syntax, see Chapter 6, "Syntax and Use of obj.conf," in *Sun Java System Web Server 7.0 Update 4 Administrator's Configuration File Reference*.

Configuring Multiple FastCGI Applications

You cannot configure multiple FastCGI applications through the Administration Console or CLI. As a workaround, you can configure multiple applications by modifying the `obj.conf` file. For example:

```
<If $uri =~ '^/fcgi/(.*)'>
Service fn="responder-fastcgi"
```

```
 app-path="/export/home/bits/fastcgi/fcgi-2.4.0/examples/$1"
 app-env="LD_LIBRARY_LIBRARY_PATH=/export/home/bits/fastcgi/fcgi-2.4.0/libfcgi/.libs"
</If>
```

The expression creates the <app-path> process, which does not need to be configured separately.

Note – You cannot configure the same `bind-path` for multiple applications because it results in startup failure due to a common `bind-path`.

Configuring the Virtual Hosting Environment

The virtual hosting environment aims to protect potential security and performance problems associated with sharing a PHP engine with several virtual servers.

Using Web Server 7.0 environment variables, you can assign same PHP binary with a separate engine bound to each virtual server. Be sure that each virtual server has its own php.ini file.

```
Service fn=responder-fastcgi
        app-path="/path/to/php/php_fcgi"
        bind-path="$(lc($urlhost))"
        req-retry=5
        type="*magnus-internal/fastcgi*"
        app-env="PHPRC=/path/to/users/$(lc($urlhost))/config"
        app-env="PHP_FCGI_CHILDREN=5"
        app-env="PHP_FCGI_MAX_REQUEST=200"
        min-procs=1
        restart-interval=10
        bucket="php-bucket"
        rlimit_cpu=60
```

The Web Server `tmp` directory now shows Unix domain sockets named after individual virtual servers processing PHP requests. This configuration is possible by using a single PHP FastCGI binary for all users. Thus, the single binary must possess all the required plugins compiled with it. The solution for the previously mentioned difficulty is to ensure that each user has a own copy of the PHP binary as needed.

```
Service fn=responder-fastcgi
        app-path="/path/to/users/$(lc($urlhost))/php_fcgi"
        bind-path="$(lc($urlhost))"
        req-retry=5
        type="*magnus-internal/fastcgi*"
        app-env="PHPRC=/path/to/users/$(lc($urlhost))/config"
        app-env="PHP_FCGI_CHILDREN=5"
        app-env="PHP_FCGI_MAX_REQUEST=200"
        min-procs=1
```

```
        restart-interval=10
        bucket="php-bucket"
        rlimit_cpu=60
```

It is also possible to allow different PHP binaries for each application by controlling the structure of the URI space.

For example:

If the URI space is structured as:

```
/app/foo.php
```

where `/app` is the name of the overall application and is always the first directory in the URI structure ending with a PHP file.

```
<If uri~=^/(\w+)/\w+\.php$>
                        Service fn=responder-fastcgi
                        app-path="/path/to/users/$(lc($urlhost))/$1/php_fcgi"
                        bind-path="$(lc($urlhost))_$1"
                        req-retry=5
                        type=+magnus-internal/fastcgi*"
                        app-env="PHPRC=/path/to/users/$(lc($urlhost))/config"
                        app-env="PHP_FCGI_CHILDREN=5"
                        app-env="PHP_FCGI_MAX_REQUEST=200"
                        min-procs=1
                        restart-interval=10
                        bucket="php-bucket"
                        rlimit_cpu=60
</If>
```

This invokes a specifically built PHP FastCGI binary which binds to a uniquely named Unix domain socket. Thus, there is no interference with another PHP application or another virtual server. However, this process uses up a lot of memory because of many PHP processes around.

Sample Configuration File

This is a sample configuration file that configures PHP with FastCGI.

```
<If -f $path>
Service type="magnus-internal/php"
    fn="responder-fastcgi"
    app-path="/opt/coolstack/php5/bin/php-cgi"
    bind-path="localhost:3101"
    app-env="PHPRC=/opt/coolstack/php5"
    app-env="PHP_FCGI_CHILDREN=5"
    app-env="PHP_FCGI_MAX_REQUEST=200"
    app-env="FCGI_WEB_SERVER_ADDRS=127.0.0.1"
```

```
    req-retry=5
    restart-interval=10
    bucket="php-bucket"

</If>
<Else>
Service type="magnus-internal/php" fn="set-variable" error="404"
</Else>
```

Troubleshooting FastCGI Plug-in

Fastcgistub is a process manager that manages the lifecycle of the FastCGI application processes. Fastcgistub logs its messages into a `Fastcgistub.log` file under Web Server's temporary directory. In case there are any errors, checking this file can help in debugging the problem.

Problem: FastCGI requests are not getting served.

Possible cause and solutions are as follows:

1. Check if the FastCGI plug-in is loaded. If the following message appears during Web Server startup, then the plug-in is loaded successfully. Otherwise, check the path to the plug-in library within `magnus.conf`:
   `FCGI1000: Sun Java System Web Server 7.0 Update 3 FastCGI NSAPI Plugin <` *build info*>
2. Check if the request mapping is correctly specified within `obj.conf`. For more information on the `obj.conf` file, see the Sun Java System Web Server *Administrator's Configuration Reference File*.
3. Check the errors log for any possible error messages.
4. Check the permissions of the stub binary and FastCGI applications. If enough permissions are not given, the plug-in fails to start the stub or the application.
5. Check the `Fastcgistub.log` file for any possible errors on the stub side. You can find the log details in `<instances>/logs`.
6. If possible, run the FastCGI application in standalone mode and check if it runs without any issues.

If any library dependency errors are thrown, specify the `LD_LIBRARY_PATH` in the `obj.conf` as `app-env` parameter with `LD_LIBRARY_PATH=`<*dependency library paths*> value .

Problem: FastCGI application is not getting started.

Possible cause and solutions are as follows:

Check `Fastcgistub.log` file for the following log messages:

```
..
<pid> process startup failure, trying to restart
...
Even after trying <n> time(s), <application path> process failed to start...no more retries
```

One of the reasons for startup failures can be the failure to load the dependent library. This issue can be resolved by specifying the appropriate library path(s) as a `app-env` parameter value to the FastCGI application configured in the obj.conf file. For example:

```
Service fn="responder_fastcgi" app-path="/fastcgi/c/tux-app" bind-path="localhost:2112"
app-env="LD_LIBRARY_PATH=/tuxedo/lib"
```

Developing FastCGI Applications

FastCGI applications can be developed using Perl, PHP, C and Java. The following sections briefly describe the procedure to develop the application using some of the popular programming languages.

▼ Executing a FastCGI Application

1. **Stop the Web Server.**
2. **Restart the Web Server.**
3. **Access the application that has "fcgi" as the application root.**

   For Example: `http://localhost/fcgi/ListDir.php`

Structure of a FastCGI Application

A typical FastCGI application has the following code structure:

```
Initialization code

Start of response loop
        body of response loop
End of response loop
```

The initialization code is run only once at the time of the application initialization. Initialization code usually performs time-consuming operations such as opening databases or calculating values for tables or bitmaps. The main task of converting a CGI program into a FastCGI program is to separate the initialization code from the code that needs to run for each request.

The response loop runs continuously, waiting for client requests to arrive. The loop starts with a call to `FCGI_Accept`, a routine in the FastCGI library. The `FCGI_Accept` routine blocks program execution until a client requests the FastCGI application. When a client request comes in, `FCGI_Accept` unblocks, runs one iteration of the response loop body, and then blocks again waiting for another client request. The loop terminates only when a System Administrator or the Web Server kills the FastCGI application.

Using Perl

Download and install the latest FCGI module from CPAN. For ActivePerl, the modules can be downloaded from `http://aspn.activestate.com/ASPN/Downloads/ActivePerl/PPM/Zips`.

For more information on writing FastCGI applications using Perl, see `http://www.fastcgi.com/#TheDevKit`

Using PHP

Beginning with PHP 4.3.0, FastCGI became a supported configuration for the PHP engine. To compile the PHP 4.3.x or greater engine with support for FastCGI, include the configure switch `--enable-fastcgi` as part of the build process, for example:

```
./configure <other-options> --enable-fastcgi
gmake
```

After compilation, the `php` binary will be FastCGI enabled.

When using PHP versions 5.1.2 or earlier (including PHP 4.x) the FastCGI plug-in should be configured with bind-path in host:port format. For example, bind-path = “localhost:3333”.

For PHP versions 5.1.3 and later, the bind-path is optional. If specified, it should not be in “host:port” format. It can be a string. For example, bind-path = “myphpbindpath”.

Using C/Java

FastCGI development kit provides APIs to write FastCGI C/Java applications. You can download the kit from `http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm`.

To build the downloaded FastCGI development kit, perform the following steps:

1. Unpack the tar file. This action creates a new directory called `fcgi-devel-kit`
2. Execute this sequence of commands in the `fcgi-devel-kit` directory:
   a. ./configure
   b. make

For more information on writing FastCGI applications using C, see `http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm#S3`

For more information on writing FastCGI applications using Java, see
`http://www.fastcgi.com/devkit/doc/fcgi-java.htm`

Configuring FastCGI Plug-in on Web Server from Administration Console

▼ To Configure FastCGI Plug-in from Administration Console

1. **Download the FastCGI enabled Sun Java System Web Server 7.0 PHP Add-On 1.0 from:**
   `http://www.sun.com/download/index.jsp`

2. **Configure PHP as a FastCGI server on Web Server.**

   a. **Unpack the** `phppack-5_2_0*.zip` **to** `/export/home`

   ```
   $ cd /export/home; unzip phppack-5_2_0*.zip
   ```

   b. **Start the Administration Server.**

   ```
   $ <webserver-install-root>/admin-server/bin/startserv
   ```

   c. **Configure the FastCGI handler using Administration Console.**

      i. **Login to the Administration Console.**

      ii. **Click Edit Virtual Server from Virtual Server Tasks.**

      iii. **Under Virtual Server General Properties, click the Content Handling tab.**

      iv. **Under Content Handling — General Properties, click the FastCGI tab.**

      v. **Click New to add a new URI with FastCGI handler mapping.**
      Enter the following values:

      - Applies To: Select **New URI** and enter `/fastcgi/*`
      - Role: Select **Responder** from the drop-down list.
      - Application Path: Enter `/export/home/php/bin/php` as the path.
      - Environment Variables: Enter the variable:

      ```
      "PHPRC=/export/home/php","LD_LIBRARY_PATH=/export/home/php",
      "LD_LIBRARY_PATH_64=/export/home/php/64"
      ```

      vi. **Click OK. Click Deploy if needed.**

3 **Create a symbolic link.**

```
$ ln -s <webserver-install-root>/samples/fastcgi <webserver-instance-docroot>
```

4 **Run the samples.**

- Hello World sample URL

  `http://<host-name>:<webserver-instance-port>/fastcgi/HelloWorld.php`

- Directory Listing sample URL

  `http://<host-name>:<webserver-instance-port>/fastcgi/directory.php`

- Page Counter sample URL

  `http://<host-name>:<webserver-instance-port>/fastcgi/pageCounter.php`

- Server Information sample URL

  `http://<host-name>:<webserver-instance-port>/fastcgi/serverinfo.php`

Configuring FastCGI Plug-in on Web Server from CLI

There are five CLI commands associated with FastCGI handler which are listed below:

- `create-fastcgi-handler(1)`
- `delete-fastcgi-handler(1)`
- `get-fastcgi-handler-prop(1)`
- `set-fastcgi-handler-prop(1)`
- `list-fastcgi-handlers(1)`

▼ To Configure FastCGI Plug-in from CLI

1 **Invoke the following command to create a FastCGI handler.**

```
wadm> create-fastcgi-handler --config=test --vs=test --uri-pattern=/php/* --role=filter
 --app-path=C:\\php\\phppack-5_2_0-windows-i586\\php\\php-cgi.exe
```

A FastCGI handler with role as `Filter` is created.

For more information, see CLI Reference, `create-fastcgi-handler(1)`

2 **Deploy the configuration by invoking the following command.**

```
wadm> deploy-config test
```

Note – If you are creating the FastCGI handler for the first time, you should restart the instance after deploying the configuration.

```
wadm> restart-instance --config=test localhost
```

Running FastCGI Enabled PHP Application in Remote Mode

You can run FastCGI enabled PHP in remote mode and configure Sun Java System Web Server. This enables the Web Server to pass requests to the remote PHP engine.

▼ To Run FastCGI Enabled PHP Application

1 **Run FastCGI enabled PHP.**

```
$ php -b <hostname>:<port> &
```

For example:

```
$ php -b localhost:4321 &
```

Note – You can check whether or not the PHP you are using is FastCGI-enabled by running the command:

```
$ php -v

PHP 5.2.5 (cgi-fcgi) (built: May  8 2008 12:50:19)
Copyright (c) 1997-2007 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies
```

In the output, look for `cgi-fcgi` to confirm.

2 **Configure the Sun Java System Web Server using CLI.**

For example, a Web Server instance named test is created.

3 **Using the CLI, execute the following command:**

```
wadm> create-fastcgi-handler --config=test --vs=test
 --uri-pattern=/php/* --role=responder --bind-path="localhost:4321"
wadm> deploy-config test
```

A FastCGI handler with the role as Responder is created.

4 **Restart your instance.**

```
wadm> restart-instance --config=test localhost
```

After completing the configuration, you can check if the requests from Web Server are being forwarded to the remote PHP engine.

a. **Place the below sample PHP script in the php subdirectory of your instance docroot, that is,** `<instance-dir>/docs_directory`

```
info.php:
<?php
phpinfo();
?>
```

b. **Access the remote PHP engine URL:** `http://localhost:<webserverport>/php/info.php` **to verify the status of your request.**

Sample FastCGI Applications

This section contains sample FastCGI applications written using PHP, Perl and C.

Responder application in PHP (ListDir.php)

```
<?php
         $dir = "/tmp/";

        // Open a known directory, and proceed to read its contents
        if (is_dir($dir)) {
            if ($dh = opendir($dir)) {
                 while (($file = readdir($dh)) !== false) {
                     echo "filename: $file : filetype: " . filetype($dir . $file) . "\n";
                 }
                closedir($dh);
            }

        }
?>
```

obj.conf snippet for the above example:

```
<Object name="default">
        NameTrans fn="assign-name" from="/fcgi/*" name="responder.fcgi"
```

```
</Object>
<Object name="responder.fcgi">
        Service fn="responder-fastcgi" app-path="/foo/fastcgi-enabled-php-installation/bin/php"
            bind-path="localhost:3431"  min-procs=3
</Object>
```

Authorizer application in Perl (SimpleAuth.pl)

```
#!/usr/bin/perl

use FCGI;

while (FCGI::accept >= 0) {
        if( $ENV{'HTTP_AUTHORIZATION'} ) {
        # This value can be further decoded to get the actual
            # username and password and then
        # perform some kind of user validation. This program only
        # checks for the presence of
        # of this environment param and is not really bothered about its value

        print( "Status: 200\r\n" );
        print( "\r\n" );

    } else {

        print( "Status: 401\r\n" );
        print( "WWW-Authenticate: basic realm=\"foo\"\r\n" );
        print( "\r\n" );

   }

}
```

obj.conf settings for the above example:

```
<Object name="responder.fcgi">
        AuthTrans fn="auth-fastcgi" app-path="/fastcgi/apps/auth/SimpleAuth.pl"
        bind-path="localhost:3432"
        Service fn="responder-fastcgi" app-path="/foo/fastcgi-enabled-php-installation/bin/php"
        bind-path="localhost:3433" app-env="PHP_FCGI_CHILDREN=8" min-procs=1
</Object>
```

On first request to `http://localhost/fcgi/php/ListDir.php`, the authentication dialogue box is displayed by the browser. After the user enters the username and password, the contents of "/tmp" directory are listed.

Filter application in C (SimpleFilter.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcgi_stdio.h>

        void main(void) {
            size_t PageSize = 1024 * 3;
            char *page;
            FCGX_Stream *in, *out, *err;
            FCGX_ParamArray envp;

             int count=0;
            page = (char *)malloc(PageSize);

             if (page == NULL) {

                printf("Content-type: text/x-server-parsed-html\r\n");
                printf("<title>malloc failure</title>");
                printf("<h1>Cannot allocate memory to run filter. exiting</h1>");
                printf("\r\n\r\n");
                exit(2);
            }

            while(FCGI_Accept() >= 0) {

             char *tmp;
             char *execcgi;
             char *dataLenStr = NULL;
             int numchars = 0;
             int stdinDataSize = 0;
             int filterDataLen = 0;
             int dataToBeRead = 0;
             int x = 0;
             int loopCount = 0;


                  count++;
             dataLenStr = getenv("FCGI_DATA_LENGTH");

             if(dataLenStr)
                 filterDataLen = atoi(dataLenStr);

                /* clear out stdin */
             while (EOF != getc(stdin)) {
                 stdinDataSize++;
             }
```

```
        dataToBeRead = filterDataLen;
    FCGI_StartFilterData();
    tmp = page; /** just in case fread or fwrite moves our pointer **/


        //start responding
        printf("Content-type: text/plain\r\n");
        printf("\r\n"); /** send a new line at the beginning **/
        printf("<title>SIMPLE FILTER</title>");
        printf(<h1>This page was Filtered by SimpleFilter FastCGI filter</h1>");
        printf("file size=%d<br>", filterDatalen);
        printf("stdin size=%d<br>, stdinDataSize);


        while(dataToBeRead > 0 ) {
           x = 0;
           page = tmp;

           if(dataToBeRead > PageSize)
              x = PageSize;
           else
               x = dataToBeRead;
            numchars = fread((void *)(page), 1, x, stdin);

            if( numchars == 0 )
                continue;
            /** at this point your data is in page pointer, so do
            whatever you want
      with it before sending it back to the server.
            In this example, no data is manipulated. Only the count of number of
      times the filter data is read and the total bytes read
            at the end of every
      loop is printed. **/

            dataToBeRead -= numchars;
            loopCount++;
            printf("loop count = %d ... so far read %d bytes <br>", loopCount,
            (filterDatalen - dataToBeRead));
        }
        printf("\r\n\r\n"); /** send a new line at the end of transfer **/

        fflush(stdout);

        page = tmp; /** restore page pointer **/
        memset(page,NULL,numchars);
}
```

```
        free(page);
}
```

Example `obj.conf` settings for the above example.

If this FastCGI application is available on the same machine where the Web Server is running, then

```
<Object name=<"filter.fcgi">
                      Service fn="filter-fastcgi" app-path="/fastcgi/apps/filter/SimpleFilter.exe"
            bind-path="localhost:3434" app-env="LD_LIBRARY_PATH=/fastcgi/fcgi-2.4/libfcgi/.libs"
</Object>
```

If the application is running on a remote machine, then the following lines of code must be included in the `obj.conf` file:

```
<Object name="filter.fcgi">
            Service fn="filter-fastcgi" bind-path="<remote-host>:<remote-port>"
</Object>
```

If "FilterThisFile" of size "26868" bytes located under the `fcgi` directory under the Web Server instance's docroot directory is the file to be filtered a request to "http://localhost/fcgi/filter/FilterThisFile" produces the following output:

```
This page was Filtered by SimpleFilter FastCGI filter


file size = 26868
stdin size = 0
loop count = 1... so far read 3072 bytes
loop count = 2... so far read 6144 bytes
loop count = 3... so far read 9216 bytes
loop count = 4... so far read 12288 bytes
loop count = 5... so far read 15360 bytes
loop count = 6... so far read 18432 bytes
loop count = 7... so far read 21504 bytes
loop count = 8... so far read 24576 bytes
loop count = 9... so far read 26868 bytes
```

◆ ◆ ◆ APPENDIX C

Web Services

To run web services on Sun Java System Web Server 7.0, no extra configuration is needed. JWSDP is integrated with the server and therefore all JWSDP web applications should run when deployed as a web application.

For more information on deploying web applications, see "Adding a Web Application" on page 177.

Running JWSDP 2.0 samples on Web Server 7.0

You need to modify the configuration files of web application samples in JWSDP 2.0 before deploying to Web Server 7.0. Specifically, configuration files in the `jaxws` samples need to be edited to make them deployable on Web Server 7.0. The steps are as follows:

▼ Running JWSDP 2.0 samples

1. **Download JWSDP 2.0.**

2. **Create a Web Server specific `sjsws.props` at $JWSDP_HOME/jwsdp-shared/bin.**

   A sample `sjws.props` is provided below. All the fields are mandatory.

```
ADMIN_USER=admin
ADMIN_PORT=8800
ADMIN_HOST=localhost
ADMIN_PASSWORD_FILE=/tmp/admin.passwd
CONFIG=jwsdp
VS=jwsdp
WS_HOME=/export/ws7.0
WS_PORT=5555
WS_HOST=localhost
```

Note – The `admin.passwd` file has the administrator' server password. An example of this entry will be: `wadm_password=adminadmin`

3 Modify configuration files.

Modify `build.xml` and `etc/deploy-targets.xml` files of the sample you plan to run. Note that the changes needed in `deploy-targets.xml` is not sample specific. You should be able to use a master copy and copy it into the etc. directory of the application you plan to run.

`build.xml` **changes**.

Comment out the Application Server `lib.home` definition at the top of the `build.xml` and add the Web Server lib location. The changed `build.xml` example is shown below:

```
<!--
   **                                                      **
   ** Comment out the Application Server lib.home declaration **
   **                                                      **
    <property file="../../../jwsdp-shared/bin/sjsas.props"/>
      <condition property="lib.home" value="${DOMAIN_DIR}/../lib">
        <available file="../../../jwsdp-shared/bin/sjsas.props"/>
      </condition>
    <condition property="lib.home" value="${env.JAXWS_HOME}/lib">
      <not>
        <available file="../../../jwsdp-shared/bin/sjsas.props"/>
      </not>
    </condition>
-->
<!--
   ** Add the Web Server library location **
-->
    <property name="lib.home" value="${WS_HOME}/lib" />
```

`deploy-targets.xml` **changes**.

Replace the `etc/deploy-targets.xml` with a web server specific `deploy-targets.xml` . This change will deploy the web application to the Web Server . An example of `deploy-targets.xml` file is shown below:

```
<property environment="env"/>
<!-- Loading Web Server properties -->
<property environment="env"/>
<property file="../../../jwsdp-shared/bin/sjsws.props"/>
<property name="ws.home" value="${WS_HOME}"/>
<property name="ws.admin" value="${ws.home}/bin/wadm"/>
<property name="lib.sample.home" value="${basedir}/../lib"/>
<property name="build.home" value="${basedir}/build"/>
<property name="build.classes.home" value="${build.home}/classes"/>
<property name="build.war.home" value="${build.home}/war"/>
```

```
<property name="config" value="${CONFIG}"/>

<target name="deploy">
    <exec executable="${ws.admin}" vmlauncher="true">
      <arg value="add-webapp" />
      <arg value="--user=${ADMIN_USER}" />
      <arg value="--password-file=${ADMIN_PASSWORD_FILE}" />
      <arg value="--host=${ADMIN_HOST}" />
      <arg value="--port=${ADMIN_PORT}" />
      <arg value="--config=${CONFIG}" />
      <arg value="--vs=${VS}" />
      <arg value="--uri=/jaxws-${ant.project.name}" />
      <arg value="${build.war.home}/jaxws-${ant.project.name}.war" />
    </exec>

    <antcall target="commit-config" />
</target>

<target name="commit-config">
    <exec executable="${ws.admin}" vmlauncher="true">
      <arg value="deploy-config" />
      <arg value="--user=${ADMIN_USER}" />
      <arg value="--password-file=${ADMIN_PASSWORD_FILE}" />
      <arg value="--host=${ADMIN_HOST}" />
      <arg value="--port=${ADMIN_PORT}" />
      <arg value="--force=true" />
      <arg value="${CONFIG}" />
    </exec>
</target>
```

◆ ◆ ◆ APPENDIX D

Windows CGI Progams

Installing Windows CGI Programs

This section discusses how to install Windows CGI Programs. 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 Java System Web Server. For more information on using the CGI, see Chapter 3, "Using Common Gateway Interface," in *Sun Java System Web Server 7.0 Update 4 Developer's Guide*

Specifying a Shell CGI Directory (Windows)

To create a directory for your shell CGI files, perform the following steps.

▼ To Create a Directory for your Shell CGI Files

1. **Create the shell directory on your computer. This directory does not have to be a subdirectory of your document root directory.**
2. **From the Home page, select the virtual server and then go to the Edit Virtual Server>Content Handling>CGI (sub tab).**
3. **In the CGI directories table, click New.**

   A new 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 Directory field, enter the absolute path to the directory you created.**

   Note – 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. **Select the CGI or Shell CGI option.**

   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.
7. **Click OK.**

Specifying Windows CGI as a File Type

You can use the Sun Java System Web Server MIME Types window to associate a file extension with the shell CGI feature. This process 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 specify a file extension for Windows CGI files, perform the following steps.

▼ To Specify a File Extension for Windows CGI files

1. **Create the shell directory on your computer. This directory does not have to be a subdirectory of your document root directory.**
2. **Go to Common Tasks > Configurations (Select configuration) > General >Mime Types (sub tab)**
3. **Click the New button to create a mime type.**

   A new window appears.
4. **Add a new mime type with the following settings:**
   - Mime Header: Select the mime header from the following `Content-type`, `Content-encoding`, `Content-language`
   - MIME Value: `magnus-internal/wincgi`.
   - 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 cannot 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 OK.**

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.

Administration Server A web-based server that contains the forms you use to configure all of your Sun Java System Web Servers.

`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 enables Glossary 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.

Granting access to an entire server or particular files and directories on it. Authorization can be restricted by criteria including hostnames and IP addresses.

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, non-forgeable, digital file issued from a third party that both communicating parties already trust.

Certificate revocation list (CRL) CA list, provided by the CA, of all revoked certificates.

certification authority (CA) An internal or third-party organization that issues digital files used for encrypted transactions.

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. Used to safeguard an unprotected server.

cipher 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 Mozilla Firefox, used to request and view World Wide Web material.

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 structure used by the server to enter information into access logs. The format is the same among all major servers, including the Sun Java System Web Server.

Compromised key list (CKL) A list of key information about users who have compromised keys. The CA also provides this list.

daemon (UNIX) A background process responsible for a particular system task.

DHCP Dynamic Host Configuration Protocol. An Internet Proposed Standard Protocol that enables a system to dynamically assign an IP to individual computers on a network.

digest authentication. enables 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.

expires header The expiration time of the returned document, specified by the remote server.

extranet An extension of a company's intranet onto the Internet, to allow customers, suppliers, and remote workers access to the data.

fancy indexing A method of indexing that provides more information than simple indexing. Fancy indexing displays a list of contents by name with file size, last modification date, and an icon reflecting file type. Because of this, fancy indexes might take longer than simple indexes for the client to load.

file extension The last part of a filename that typically defines the type of file. For example, in the filename `index.html` the file extension is `html`.

file type The format of a given file. For example, a graphics file doesn't have the same file type as a text file. File types are usually identified by the file extension (`.gif` or `.html`).

firewall A network configuration, usually both hardware and software, that protects networked computers within an organization from outside access. Firewalls are commonly used to protect information such as a network's email and data files within a physical building or organization site.

flexible log format A format used by the server for entering information into the access logs.

FORTEZZA An encryption system used by U.S. government agencies to manage sensitive but unclassified information.

FTP File Transfer Protocol. An Internet protocol that enables 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 Mozilla Firefox 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 Listener The combination of port number and IP address. Connections between the server and clients happen on an HTTP Listener.

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.

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. 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.

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.

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.

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.

`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 difficult 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.
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 web servers. An NMS is usually a powerful workstation with one or more network management applications installed.
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.
NNTP	Network News Transfer Protocol for newsgroups. You must define your news server host to use agent services on your server.
`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). The Sun Java System 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.
private key	The decryption key used in public-key encryption.
protocol	A set of rules that describes how devices on a network exchange information.
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.
public key	The encryption key used in public-key encryption.
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 enables you to extend and/or customize the core functionality of Sun Java System Web Servers and provide a scalable, efficient mechanism for building interfaces between the HTTP server and back-end applications. Also known as NSAPI.

server root A directory on the server machine dedicated to holding the server program, configuration, maintenance, and information files.

simple index The opposite of fancy indexing—this type of directory listing displays only the names of the files without any graphical elements.

SNMP Simple Network Management Protocol.

SOCKS Firewall software that establishes a connection from inside a firewall to the outside when direct connection will 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 use in a search. 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 Java System Web Server Administration Console A Java application that provides server administrators with a graphical interface for managing all Sun Java System Web Servers from one central location anywhere within an enterprise network. From any installed instance of the Sun Java System Web Server Administration Console, you can see and access all the Sun Java System servers on an 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 can 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 Virtual servers are a way of setting up multiple domain names, IP addresses, and server monitoring capabilities with a single installed server.

virtual server class A collection of virtual servers that shares the same basic configuration information in a obj.conf file.

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.
Windows CGI (Windows)	CGI programs written in a Windows-based programming language such as Visual Basic.

Index

D

E

F

M

N

O

P

R

S

T

U

V

W

X