



# Sun Java System Web Server 6.1 SP7 Getting Started Guide



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

Part No: 820-0256

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux États-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivés du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des États-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

# Contents

---

<b>Preface</b> .....	5
<b>1 Basic Tasks</b> .....	11
Web Server Instances .....	11
Start the Administration Server .....	12
Accessing the Administration Server .....	13
Start the Web Server Instance .....	14
Access the Web Server .....	15
<b>2 Creating and Using Virtual Servers</b> .....	17
Using Virtual Server .....	17
▼ To create a listen socket .....	19
▼ To create a virtual server class .....	20
▼ To create a virtual server .....	21
▼ To create an additional document directory .....	23
▼ To change the document root of a virtual server .....	25
<b>3 Enabling Access Control</b> .....	27
Setting up Native Access Control .....	27
▼ To create a directory service .....	28
▼ To add a user .....	30
▼ To specify a directory service for a virtual server .....	31
▼ To set access control .....	33
<b>4 Publishing Content</b> .....	39
Set Up the Server to Handle WebDAV .....	39
▼ To enable WebDAV .....	39

▼ To enable WebDAV for the virtual server .....	40
▼ To create a WebDAV collection .....	42
Set Up a Client to Use WebDAV .....	45
<b>5 Making Content Searchable .....</b>	<b>47</b>
Create a Search Collection .....	47
Update Your Collection .....	49
Enable the Search Application .....	50
Customize the Search Application .....	51
<b>6 Running Java Web Applications On Your Server .....</b>	<b>53</b>
Deploying Web Application .....	53
▼ To deploy web application .....	53
<b>7 Setting Up Database Connectivity .....</b>	<b>57</b>
JDBC Settings .....	57
Install and Configure the JDBC driver .....	57
▼ To add PointBase Type 4 JDBC Driver to Classpath of the Web Server .....	58
▼ To start the database server .....	58
Create a Database and Table .....	59
▼ To define the JDBC connection pool .....	59
▼ To define the JDBC resource .....	62
<b>8 Using CGI .....</b>	<b>65</b>
Configuring Server to use CGI .....	66
▼ To configure your server to use CGI .....	66
<b>9 Using Sun Java System Active Server Pages .....</b>	<b>69</b>
Administering ASPs .....	69
▼ To deploy a simple application .....	70
<b>Index .....</b>	<b>73</b>

# Preface

---

The Sun Java System Web Server Getting Started guide is recommended for first time users of the Sun Java™ System Web Server 6.1 product. This guide introduces you to the basic features of the server, through hands-on exercises.

This chapter assumes that you have already installed the server and are ready to explore its features.

This preface contains information about the following topics:

- “Using the Documentation” on page 5
- “How This Guide Is Organized” on page 7
- “Documentation Conventions” on page 8
- “Product Support” on page 9

## Using the Documentation

The Sun Java System Web Server manuals are available as online files in PDF and HTML formats at <http://docs.sun.com/app/docs/coll/1308.4>.

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

TABLE P-1 Sun Java System Web Server Documentation Roadmap

For Information About	See the Following
Late-breaking information about the software and documentation	<i>Release Notes</i>
Getting started with Sun Java System Web Server, including hands-on exercises that introduce server basics and features (recommended for first-time users)	<i>Getting Started Guide</i>

TABLE P-1 Sun Java System Web Server Documentation Roadmap (Continued)

For Information About	See the Following
Performing installation and migration tasks: <ul style="list-style-type: none"> <li>■ Installing Sun Java System Web Server and its various components, supported platforms, and environments</li> <li>■ Migrating from Sun Java System Web Server 4.1 or 6.0 to Sun Java System Web Server 6.1</li> </ul>	<i>Installation and Migration Guide</i>  <b>Note:</b> If you have the Sun Java Enterprise System 1 installed on your system and you want to upgrade the Sun Java System Web Server 6.1 that is part of Sun Java Enterprise System 1 to Sun Java System Web Server 6.1 SP7, you must use the Java Enterprise System (JES) installer to perform the upgrade. Do not use the separate component installer included with Sun Java System Web Server 6.1 SP7.
Performing the following administration tasks: <ul style="list-style-type: none"> <li>■ Using the Administration and command-line interfaces</li> <li>■ Configuring server preferences</li> <li>■ Using server instances</li> <li>■ Monitoring and logging server activity</li> <li>■ Using certificates and public key cryptography to secure the server</li> <li>■ Configuring access control to secure the server</li> <li>■ Using Java 2 Platform, Standard Edition (J2SE) security features</li> <li>■ Deploying applications</li> <li>■ Managing virtual servers</li> <li>■ Defining server workload and sizing the system to meet performance needs</li> <li>■ Searching the contents and attributes of server documents, and creating a text search interface</li> <li>■ Configuring the server for content compression</li> <li>■ Configuring the server for web publishing and content authoring using WebDAV</li> </ul>	<i>Administrator's Guide</i>
Using programming technologies and APIs to do the following: <ul style="list-style-type: none"> <li>■ Extend and modify Sun Java System Web Server</li> <li>■ Dynamically generate content in response to client requests</li> <li>■ Modify the content of the server</li> </ul>	<i>Programmer's Guide</i>

TABLE P-1 Sun Java System Web Server Documentation Roadmap (Continued)

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

## How This Guide Is Organized

This guide has the following sections:

- [Chapter 1, Basic Tasks](#)  
Introduces you to a few basic tasks associated with Sun Java System Web Server, including starting and accessing the server.
- [Chapter 2, Creating and Using Virtual Servers](#)  
Demonstrates the creation and use of listen sockets, virtual server classes, and virtual servers.
- [Chapter 3, Enabling Access Control](#)  
Shows how you can configure access control to protect resources on a virtual server.
- [Chapter 4, Publishing Content](#)  
Demonstrates how to set up a virtual server to support web publishing through Web-based Distributed Authoring and Versioning (WebDAV) and also, how to access resources using a WebDAV-enabled client.
- [Chapter 5, Making Content Searchable](#)  
Demonstrates how you can use the search functionality to make content searchable on a virtual server.
- [Chapter 6, Running Java Web Applications On Your Server](#)  
Uses a sample application to show how you can deploy a Java web application on a virtual server.
- [Chapter 7, Setting Up Database Connectivity](#)  
Shows how you can set up database connectivity on a virtual server and create J2SE resources to store and retrieve data from a database.
- [Chapter 8, Using CGI](#)

Demonstrates how to enable a virtual server to handle dynamic content using Common Gateway Interface (CGI) programs.

- [Chapter 9, Using Sun Java System Active Server Pages](#)

Demonstrates how to enable a virtual server to handle dynamic content using Sun™ Java System Active Server Pages.

## Documentation Conventions

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

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

- **URLs** are given in the format:

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

In these URLs, *server* is the server name where applications are run; *domain* is your Internet domain name; *path* is the server's directory structure; and *file* is an individual filename. Italic items in URLs are placeholders.

- **Font conventions** include:

- The monospace font is used for sample code and code listings, API and language elements (such as function names and class names), file names, pathnames, directory names, and HTML tags.
- *Italic* type is used for code variables.
- *Italic* type is also used for book titles, emphasis, variables and placeholders, and words used in the literal sense.
- **Bold** type is used as either a paragraph lead-in or to indicate words used in the literal sense.

**Installation root directories** are indicated by *install\_dir* in this document.

By default, the location of *install\_dir* on UNIX-based platforms is:

```
/opt/SUNWwbsvr/
```

On Windows, it is:

```
C:\Sun\WebServer6.1
```



## Product Support

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

- The online support web site at:  
<http://www.sun.com/training/>
- The telephone dispatch number associated with your maintenance contract

Please have the following information available prior to contacting support. This helps to ensure that our support staff can best assist you in resolving problems:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps



◆ ◆ ◆    1  
CHAPTER 1

# Basic Tasks

---

The Sun Java System Web Server is a secure and highly available server that you can use to host web sites. Each web site you create on Sun Java System Web Server is identified by a unique URL or address. A URL is a combination of a specific IP address, a port and a host name that follows the pattern: `http://www.sun.com/`.

Sun Java System Web Server listens on the network for requests from clients (browsers) for specific pages that it hosts. The pages that the Sun Java System Web Server makes available are stored in a directory structure within its primary document directory, also referred to as the docroot, in this guide.

## Web Server Instances

When you install the Sun Java System Web Server 6.1 product, two server instances are installed by default, as listed and shown below:

- an Administration Server Instance
- a Web Server Instance

---

**Note** – If you have the Sun Java™ Enterprise System 1 installed on your system and you want to upgrade the Sun Java System Web Server 6.1 that is part of Sun Java Enterprise System 1 to Sun Java System Web Server 6.1 SP7, you must use the Java Enterprise System (JES) installer to perform the upgrade. Do not use the separate component installer included with Sun Java System Web Server 6.1 SP7.

---

The Administration Server is created in the `server_root/https-admserv` directory and you

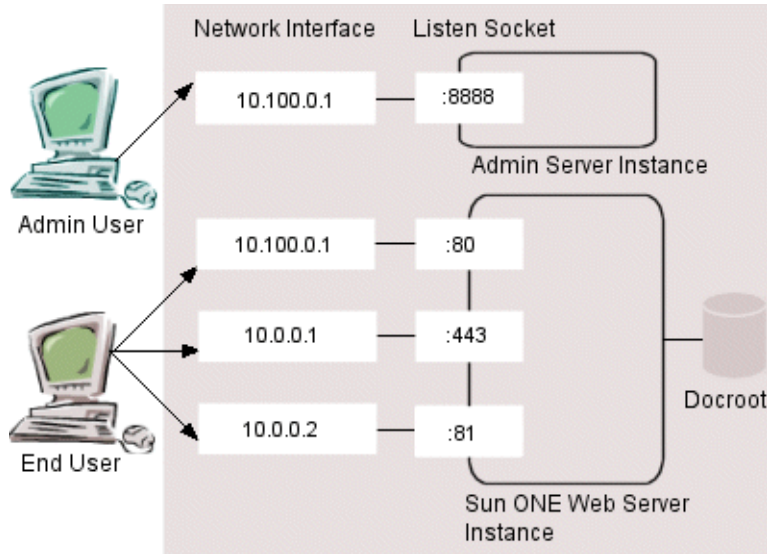


FIGURE 1-1 Web Server Instances

can use it to administer all of your Sun Java System Web Server instances, including the default instance.

The default instance is created in the `server_root/https-host.domain` directory. You specified the *host.domain* part of the directory name during installation.

This chapter includes the following sections:

- “Start the Administration Server” on page 12
- “Accessing the Administration Server” on page 13
- “Start the Web Server Instance” on page 14
- “Access the Web Server” on page 15

## Start the Administration Server

Install Sun Java System Web Server at the default directory (`/opt/SUNWwbsvr/`). To access the Administration Server at the default directory, you must first start it.

- To start the Administration Server on a UNIX machine
  - Go to `/opt/SUNWwbsvr/https-admserv`
  - Type `./start`

Alternatively, you can type `./startconsole` in the server root directory `/opt/SUNWwbsvr/`.

On Windows, the installation process creates a program group with an Start Web Server Administration Server icon. When you click this icon, the Administration Server is launched in a browser window.

## Accessing the Administration Server

After you have started the Administration Server, you can access it by specifying the Administration Server address in a browser window:

`http://server_name:administration_port`.

For Instance, if the server name specified during installation is `acme` and the default port `8888` as the port for the Administration Server to run on, open a browser and type `http://acme:8888`.

The Administration Server prompts you to authenticate yourself through a login screen. Type your username and password (you specified these when you installed the server). Once authentication is complete, you are ready to get started.

You can manage your Web Server, using the following user interfaces:

- Administration Server: allows you to manage all your servers
- Server Manager: allows you to create and manage a specific server instance
- Class Manager: allows you to create and manage a group of virtual servers
- Virtual Server Manager: allows you to create and manage a particular virtual server

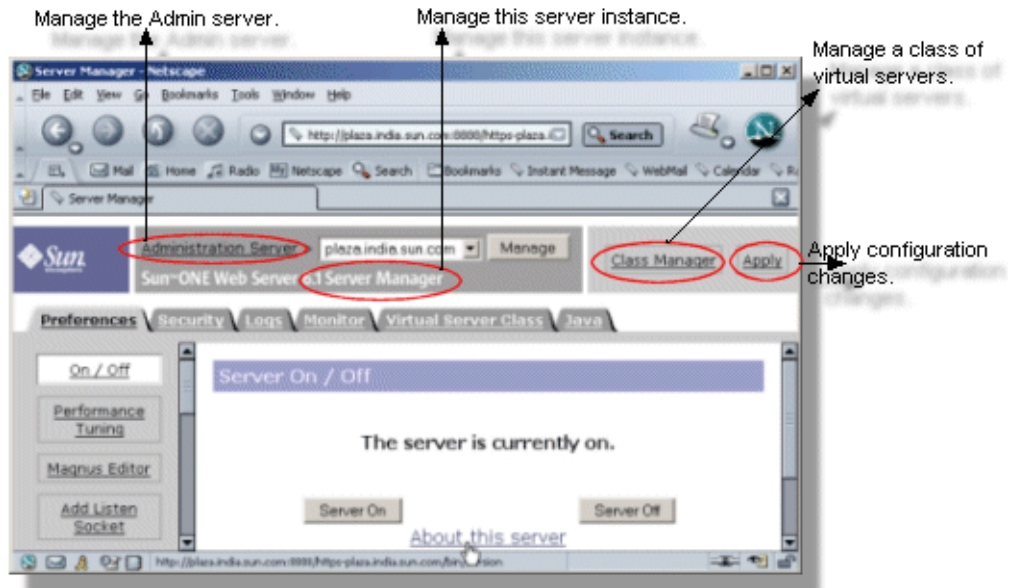


FIGURE 1-2 Administration Server Interface

For a complete description of the features each of these interfaces provides, see the *Sun Java System Web Server 6.1 SP7 Administrator's Guide*.

The default Web Server instance is turned off when you first install the product and you need to start it.

## Start the Web Server Instance

To start the default web server instance, you can use either the Administration Server interface or the start script available in the `server_root/https-host.domain` directory.

- Access the Administration Server interface in a browser and then click the Manage Server link. This brings up the Manage Server page where you can see your web server listed at the top of a drop-down menu. Click the Manage button. This takes you to the On/Off page in the Server Manager interface where you click the Server On button to start the server instance.
- Alternatively, go to `/opt/SUNWwbsvr/https-host.domain` directory, and type `./start` at the command prompt.

## Access the Web Server

To access Web Server's default home page, open a browser and type `http://acme`

The default home page of the server is displayed from the `server_root/docs` directory.

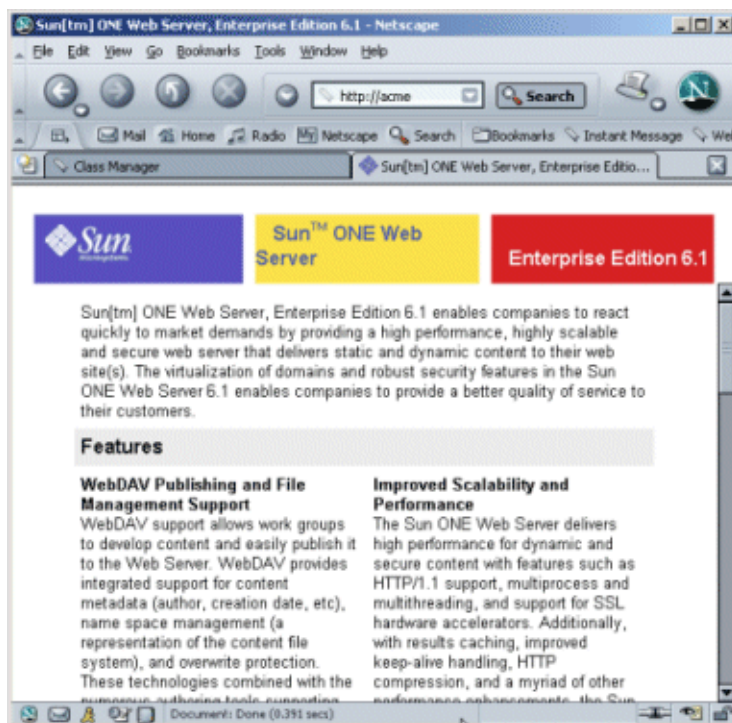


FIGURE 1-3 Web Server's Default Home Page

You can publish additional content on the site by copying the content onto the docroot directory, `server_root/docs`, of the web server instance.

The `server_root/docs` directory contains a set of default pages, including one named `index.html` (shown in the figure above). You can replace these default pages with other content; in any way. For instance, if you want to publish your own set of HTML pages, copy these into the `server_root/docs/` directory. By default, Sun Java System Web Server displays `index.html` as the server home page, which you can launch by typing `http://acme` in a browser window.

To publish additional content, you need not copy everything that you want to publish on your site into the docroot directory. You can also put content into any directory that you have mapped as an additional document directory.

This is discussed in the section [Chapter 2, Creating and Using Virtual Servers](#).





# Creating and Using Virtual Servers

---

A virtual server is a server that uses a unique combination of IP address, port number and host name to identify it. You might have several virtual servers, all of which use the same IP address and port number but are distinguished by their unique host names.

For instance, you might have one virtual server called `hr.acme.com`, and another called `mis.acme.com`, both of which reside on the same Web server instance, and listen for requests on the same port.

## Using Virtual Server

Virtual servers have the following benefits:

- **Minimize memory and hardware utilization.** Each virtual server can, for example, have individual bandwidth or connection limits specified, ensuring that no one virtual server uses more than its share of system resources.
- **Provide differentiated services to different sets of users.** For example, you could set up one virtual server to run CGI programs and another to run web applications.

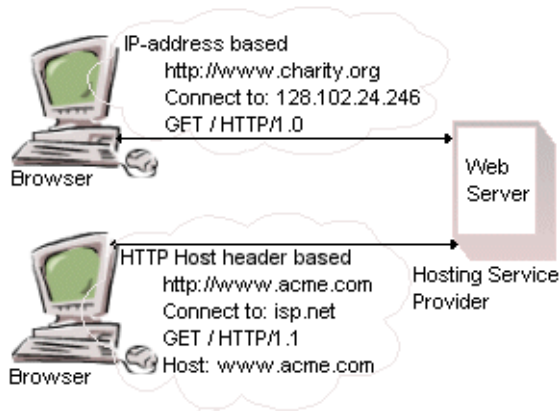


FIGURE 2-1 Types of Virtual Server

As the above figure shows, there are two different types of virtual servers. Virtual servers that use a unique IP address and port combination are known as **hardware virtual servers**. Each web site configured on a hardware virtual server has a single IP address. This means that for each web site that you add, you need a new IP address. Because IP addresses are limited, this places a constraint on the number of virtual servers that you can configure.

The other type of virtual server is the **software virtual server**. Software virtual servers use an IP address and port combination, and also the contents of an HTTP Host Header (the requested host name) to distinguish one virtual server from another. This enables one machine (with one IP address) to support multiple web sites, each of which is uniquely identified on that machine by its HTTP Host Header. This eliminates the need for many IP addresses.

Sun Java System Web Server 6.1 supports both hardware (IP address-based) and software (HTTP Host Header-based) virtual servers.

When you install Sun Java System Web Server, a default virtual server is created automatically. You can however create other virtual servers and customize them according to your specific needs.

You might have several virtual servers running on a single machine and receive requests for information from hundreds of clients. How would the server know which virtual server should handle which request? Before the server can process a request, it must accept the request via a listen socket, then direct the request to the correct virtual server, based on the value of the IP address or the HTTP Host header. You could manage virtual servers separately or group them together with other “like” servers into a virtual server class.

Before you can create a virtual server, you need two things for it to work:

- a listen socket
- a virtual server class

To create a listen socket and virtual server perform the following tasks:

- “To create a listen socket” on page 19
- “To create a virtual server class” on page 20
- “To create a virtual server” on page 21
- “To create an additional document directory” on page 23
- “To change the document root of a virtual server” on page 25

For example, if you want to create a virtual server, `hr.acme.com`, you have to decide which listen socket it would use. You could use the default listen socket, `ls1`, that’s created automatically when you install the server or you might want to add a new one. See the below example to create a new listen socket .

## ▼ To create a listen socket

- 1 Go to the Preferences tab on the Server Manager.
- 2 Click the Add Listen Socket link.
- 3 Specify the ID of the listen socket and the port number it will listen on. Retain the default values listed for the other parameters on the page.

This is shown in the [Figure 2-2: Adding Listen Socket](#):

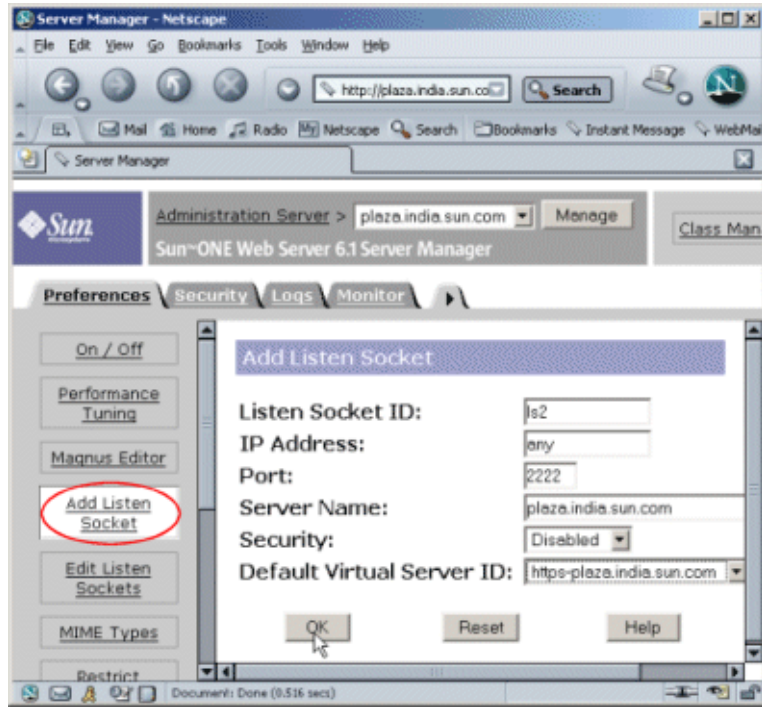


FIGURE 2-2 Adding Listen Socket

- 4 Click OK and then Apply to save and apply your changes.

Next, you need to decide which class you want your virtual server to belong to. Using classes, you can configure similar virtual servers at the same time, so you do not have to configure each one separately. You can use the default virtual server class, `vsclass1`, that is created automatically when you install the server or you can add a new class.

See the below example to create a new virtual server class, `vsclass2`.

## ▼ To create a virtual server class

- 1 Access the Virtual Server Class tab in the Server Manager.
- 2 Click the Add Class link.
- 3 Specify the virtual server class name and the document root as an absolute path for the class.

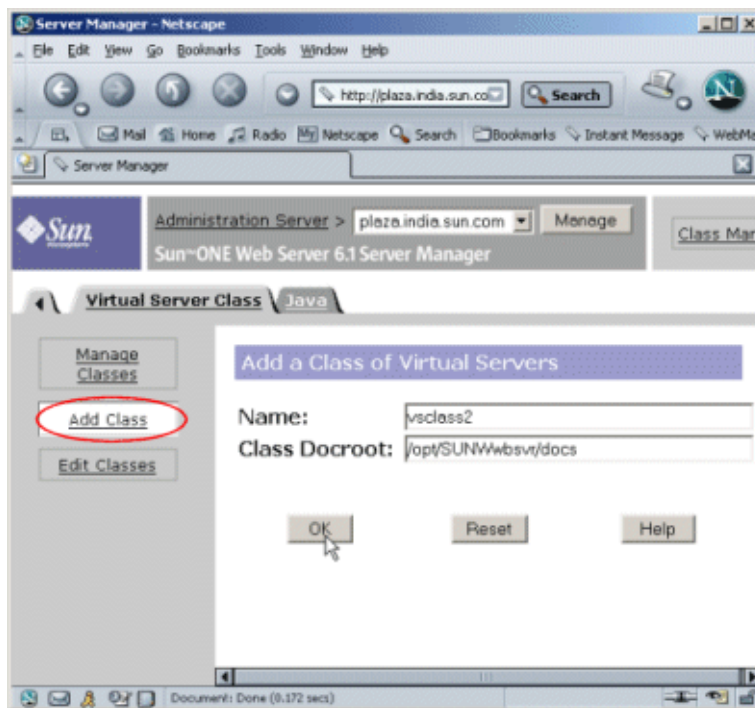


FIGURE 2-3 Adding a Class of Virtual Servers

- 4 Click OK and then Apply to save and apply your changes.

Next, create a new virtual server that will use the new listen socket we created, `ls2`, and will be managed by the virtual server class, `vsclass2`.

## ▼ To create a virtual server

- 1 Access the Virtual Server class tab in the Class Manager.
- 2 Click the Add Virtual Server link.
- 3 Specify the name of the virtual server, its connections, and the URL Hosts. In this example, we specify `acme` as the URL which clients will use to refer to the server.

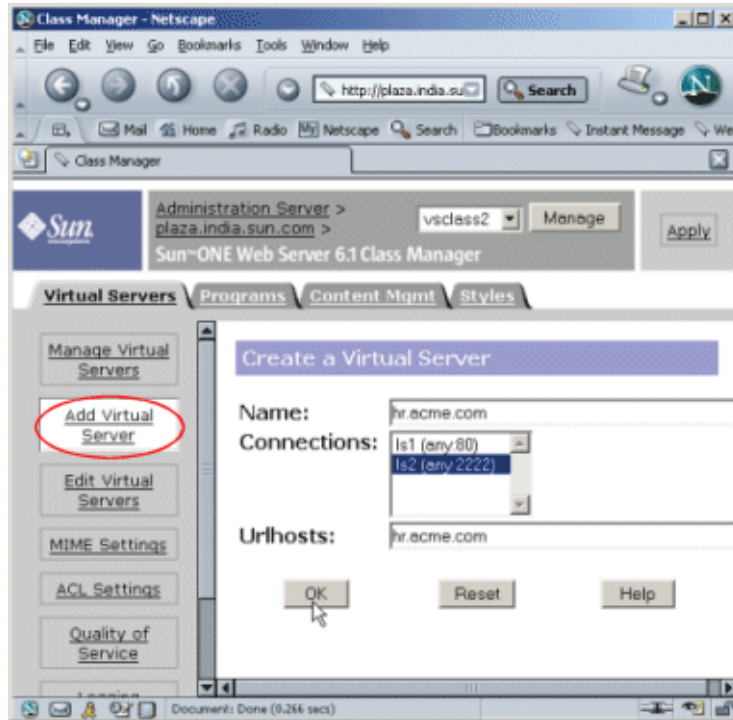


FIGURE 2-4 Creating a Virtual Server

**4 Click OK and then Apply to save and apply your changes.**

You can configure additional virtual server settings in two ways:

- Using the Class Manager
- Using the Virtual Server Manager

On the Class Manager, the pages are organized by the kind of setting you want to change. Use the Class Manager if you want certain settings to apply to all the virtual servers in the class.

On the Virtual Server Manager, the pages only pertain to one virtual server, so you can see and change all the settings for a specific server.

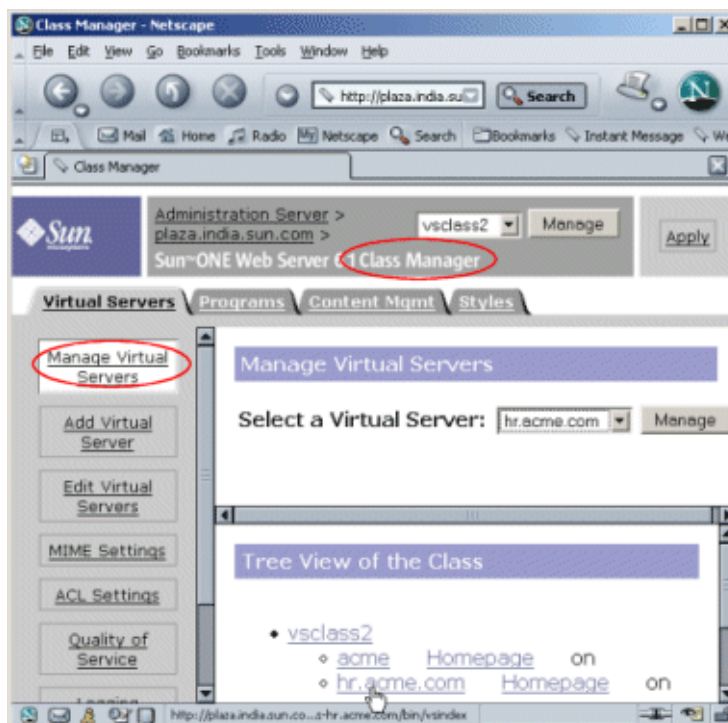


FIGURE 2-5 Managing Virtual Servers

The simplest way to publish content on the site is to put it in the docroot of the server. However, you do not have to copy everything that you want to publish on your site into the docroot directory. You can also put content into any directory that you have mapped as an additional document directory.

## ▼ To create an additional document directory

For instance, you have a folder called `/hr/publish` containing pages you want to post to your site.

- 1 Access the Virtual Server Class Manager, and click the Content Management tab.
- 2 Click Additional Document Directories.
- 3 Choose `/hr` as the URL prefix to map.

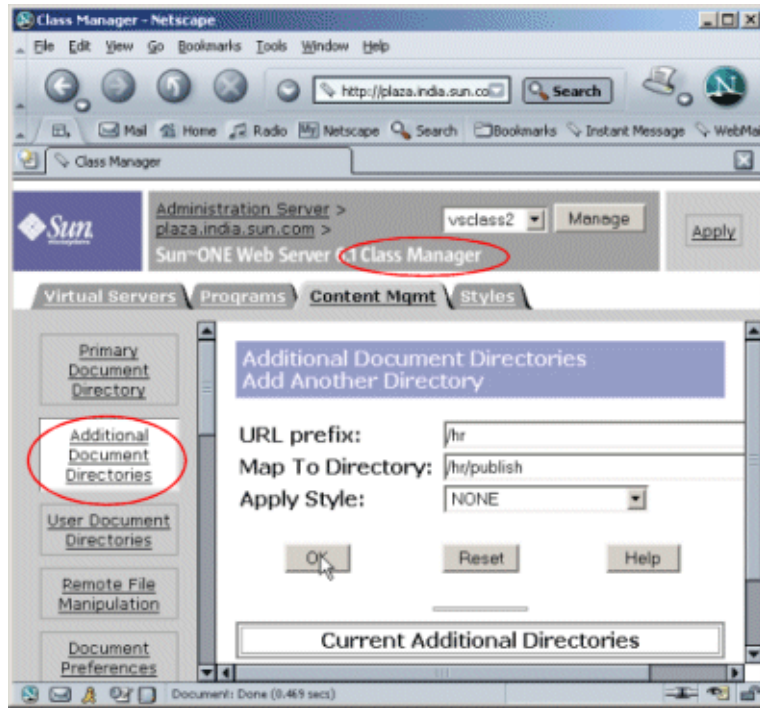


FIGURE 2-6 Additional Document Directories

- 4 **Type /hr/publish as the filesystem directory to map those URLs to.**
- 5 **Click OK.**

You can now type `http://acme/hr` in a browser window to access the content that you have mapped.

All the virtual servers in a virtual server class, by default, have the same document root.

For instance, you have two virtual servers `hr.acme.com` and `mis.acme.com` belonging to the same virtual server class. When you type `http://hr.acme.com` on a browser, you probably want them to be taken to a docroot that contains HR information, say `/acme/hr`. Similarly, when you type `http://mis.acme.com` on a browser, you want them to go to the MIS site, which has its docroot at say, `/acme/mis`. You need to configure the docroot for each virtual server individually to achieve this.



## ▼ To change the document root of a virtual server

- 1 Access the virtual server manager interface of the virtual server you want to configure, say acme.
- 2 Click the Settings link.

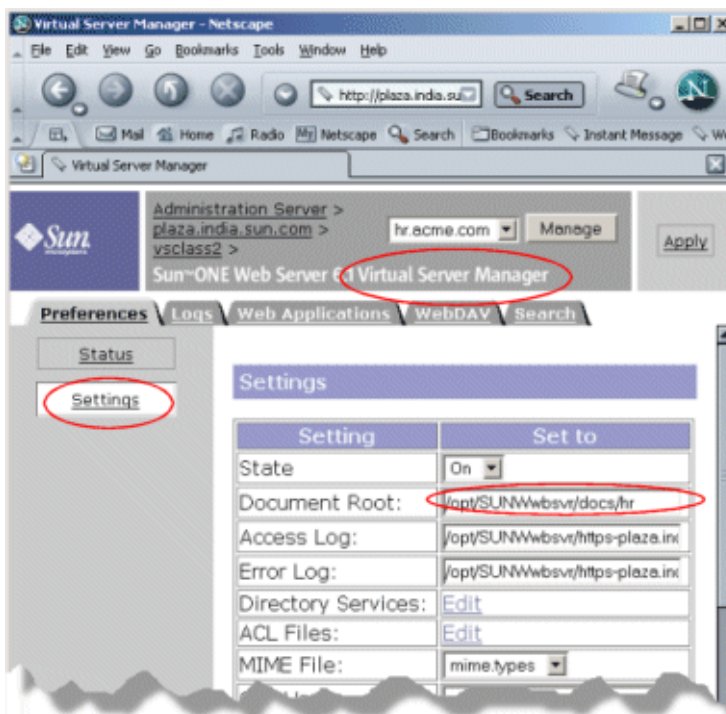


FIGURE 2-7 Virtual Server Document Root

- 3 Edit the path that appears next to the Document Root field setting to point to a new document root.
- 4 Click OK.

You can now configure your virtual servers in many other ways to provide different services to different sets of end users, based on specific needs. The rest of this guide explains some of the more common tasks:

- Chapter 3, Enabling Access Control
- Chapter 4, Publishing Content
- Chapter 5, Making Content Searchable
- Chapter 6, Running Java Web Applications On Your Server
- Chapter 7, Setting Up Database Connectivity

- [Chapter 8, Using CGI](#)
- [Chapter 9, Using Sun Java System Active Server Pages](#)

## Enabling Access Control

---

Every virtual server hosts one or more web sites. By default, you can access all the content on each web site. Sometimes, such unrestricted access might not be desirable and you might want to protect parts of your web site from unauthorized access. You can do so by setting up access control on your server.

Access control is a way of specifying who can access resources such as directories and files on your server, and what access they can have. You can allow or deny access to specified users and groups using ACLs (Access Control Lists).

There are two types of access control:

- *Host-based access control.* This enables you to control access according to specific computers. You can identify the computer by either IP address or host name. Clients accessing your site from a permitted computer are allowed access to resources.
- *User-based access control.* This offers more security than computer-based access because clients must supply a valid username and password before they can access your site. If you want to use user-based access control, you need to set up user and group accounts first.

### Setting up Native Access Control

This section describes basic tasks associated with setting up native access control. Native access control provides authenticated access for both Java-based and non-Java applications. However, if you plan to deploy Java web applications, you can leverage the benefits of Java-based security realms. The various aspects of Java security constraints are outside the scope of this guide but are discussed in detail in the Sun Java System Web Server 6.1 *Programmer's Guide to Web Applications*.

For more information on the various aspects of Java security, see *Sun Java System Web Server 6.1 SP7 Programmer's Guide to Web Applications*.

Here is a simple exercise that illustrate how a native ACL is configured. To use a combination of host-based and user-based access control perform the following tasks:

- “To create a directory service” on page 28
- “To add a user” on page 30
- “To specify a directory service for a virtual server” on page 31
- “To set access control” on page 33

Consider that you want to allow access to all files under `/hr/publish/manager` to a user named “manager” at Acme Corp. To use user-based access control (in addition to host-based access control), create a **directory service**

Since you use you first need to .

## ▼ To create a directory service

A directory service allows you to authenticate and authorize users and groups. You can configure a directory service in one of the following ways:

- Set up user information in a file
- Obtain user information from an LDAP server

In this example, we will set up user information in a file.

- 1 Access the Administration Server and choose the Global Settings tab.**
- 2 Click the Configure Directory Service link.**
- 3 From the Create New Service of Type drop-down list, choose Key File as the type of directory service. This is a text file that contains the user’s password and the list of groups to which the user belongs.**

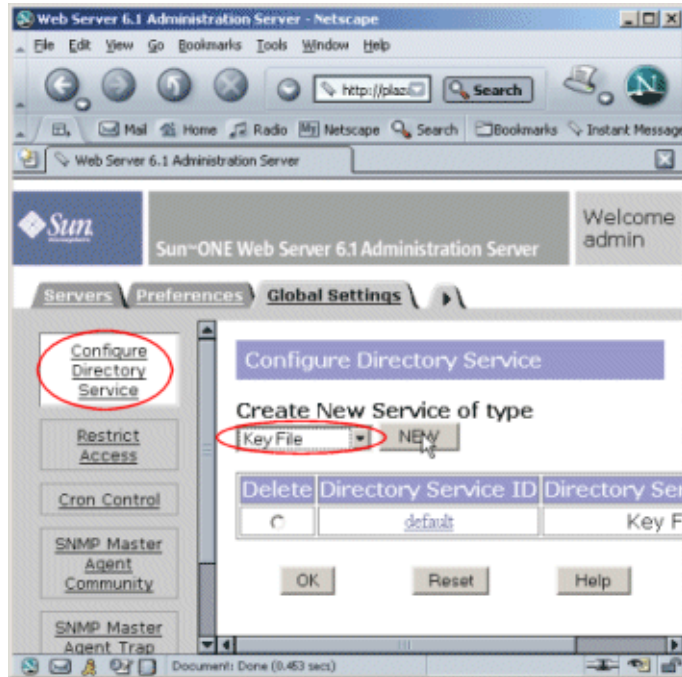


FIGURE 3-1 Creating New Directory Service Type

- 4 Click New.
- 5 Specify `keyfile1` as the Directory Service ID and `HRAuthFile` as the name of the file, as shown below:

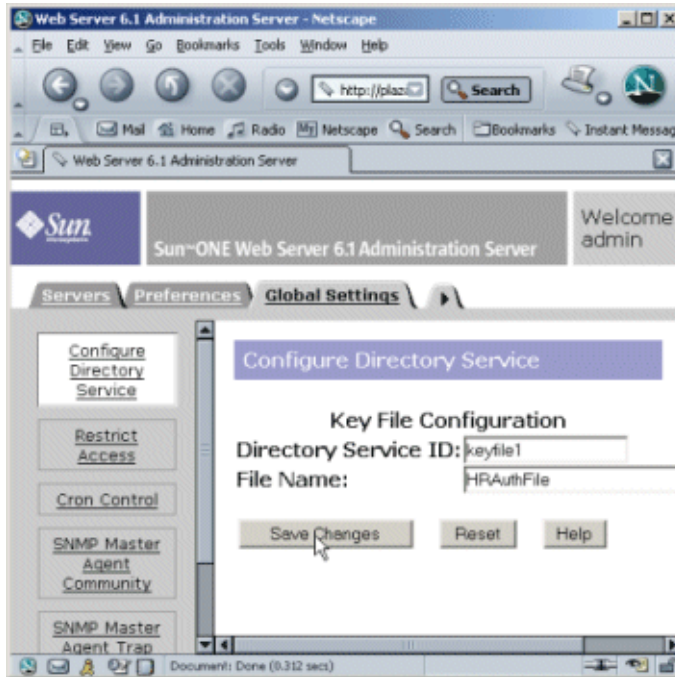


FIGURE 3-2 Configuring Directory Service Type

- 6 Click Save Changes.
- 7 Restart the server for the changes to take effect.

## ▼ To add a user

We start with creating a user ID called “manager.” This will represent the person who needs to have access to all the files in `/hr/publish/manager`.

- 1 Access the Administration Server and choose the Users & Groups tab.
- 2 Click New User.
- 3 Select “HRAuthFile” from the Select Directory service drop-down list and click Select.
- 4 Enter the required information, as shown below:

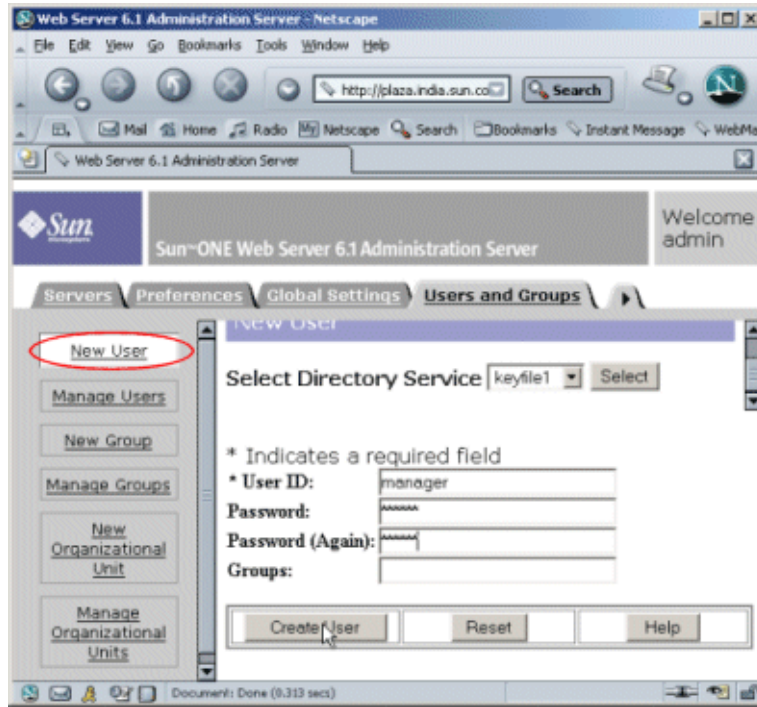


FIGURE 3-3 Creating a New user

5 Click **Create User**.

A new user is added to the file: `HRAuthFile`.

Next, you need to associate our virtual server, `hr.acme.com`, with the directory service we've created.

## ▼ To specify a directory service for a virtual server

- 1 Access the Virtual Server Manager and click the Settings link to display the Settings page for the virtual server, `hr.acme.com`.

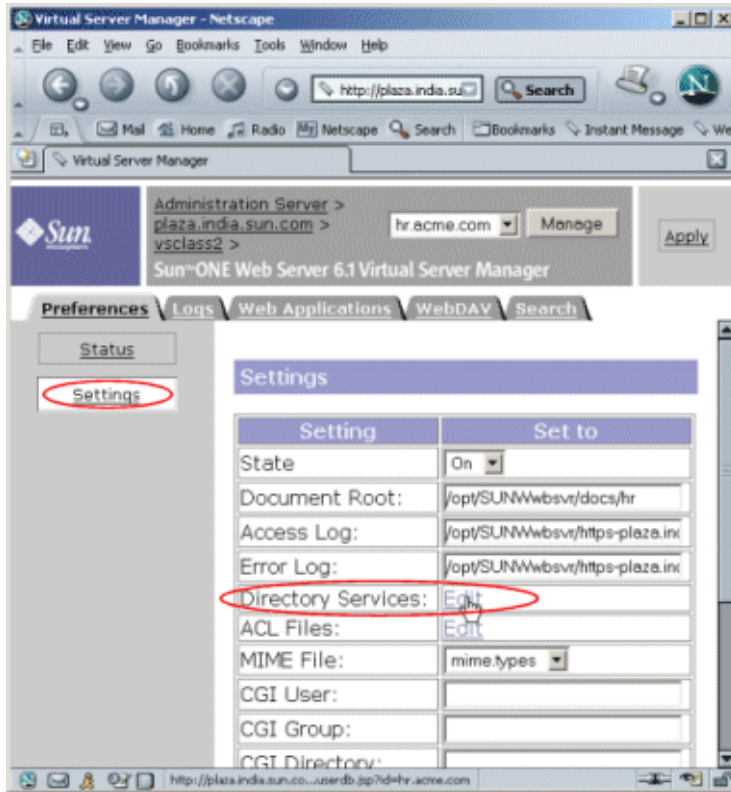


FIGURE 3-4 Specifying Directory Service

- 2 Click the Edit link next to the Directory Services setting.
- 3 Select `keyfile1` in the Pick Directory Services for Virtual Server page as shown below:



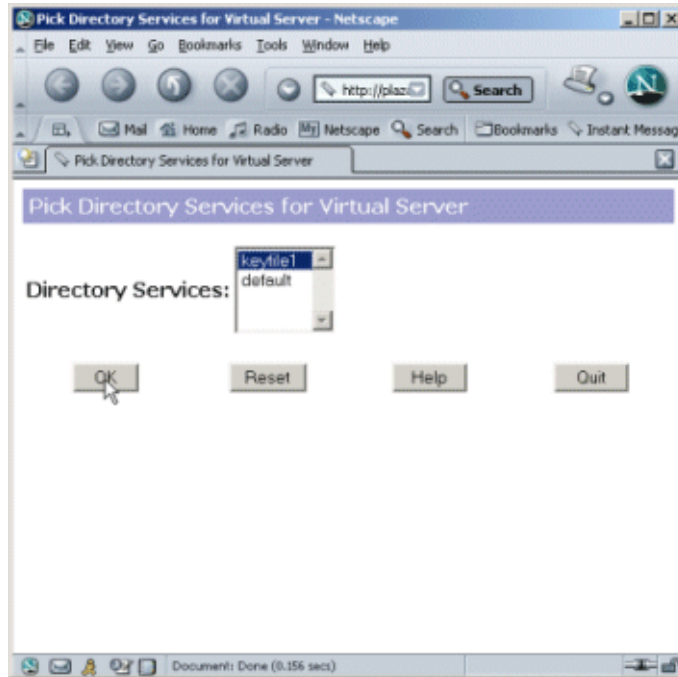


FIGURE 3-5 Editing Directory Service

- 4 **Click OK and then Apply to save and apply your changes.**

Now you can specify the required access control rules.

## ▼ To set access control

Start by creating an ACL for the virtual server `hr.acme.com`.

- 1 **Access the Server Manager and choose the Preferences tab.**
- 2 **Click the Restrict Access link.**
- 3 **Under the Option column, select the ACL file. We choose the default file in our example and click OK.**

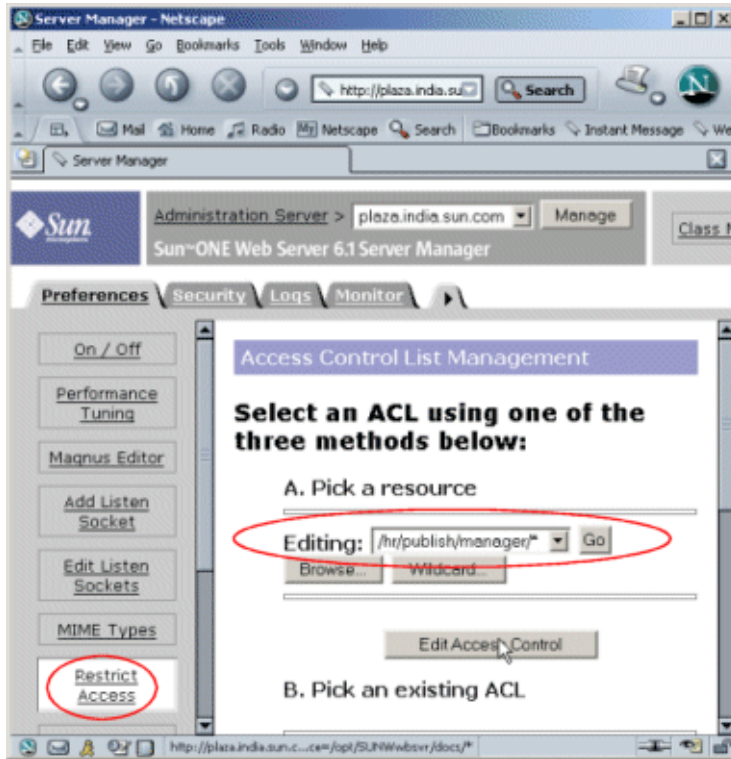


FIGURE 3-6 Access Control List Management Page

- 4 Select Pick a Resource and then specify the following wildcard pattern to control access to all files in the /hr/publish/manager directory:  
/hr/publish/manager/\*
- 5 Click the Edit Access Control button.
- 6 Check the “Access control is on” checkbox and click the New Line button.

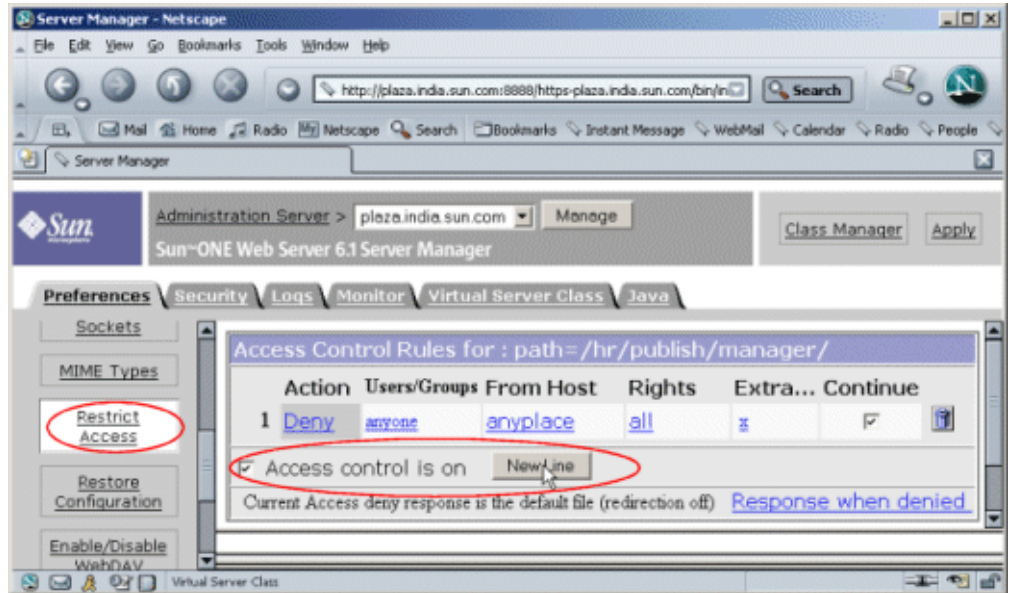


FIGURE 3-7 Editing the Access Control

- 7 Click on Deny in the second row of the Action column.

This opens the Allow /Deny page in the lower frame of [Figure 3-8: Restricting Access Control](#)

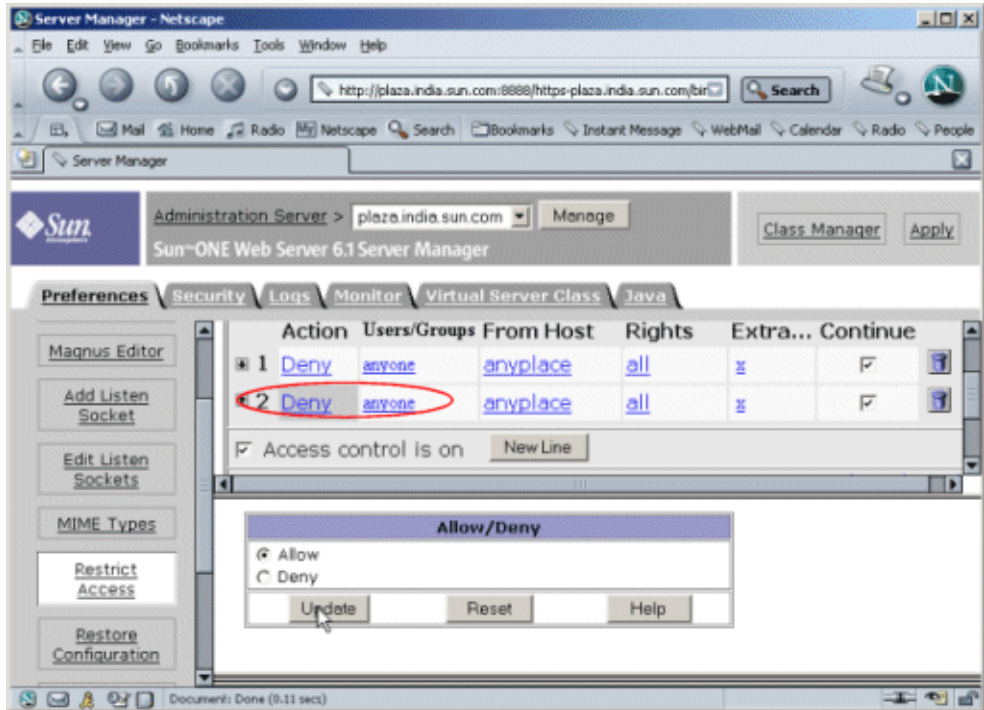


FIGURE 3-8 Restricting Access Control

- 8 Click Allow, if not selected by default, and then click Update.
- 9 Click on “anyone” in the Users/Groups column, in the top frame.
- 10 Enter “manager” as the user you will allow access to, and `keyfile1` as the authentication database, in the User/Group page that appears in the lower frame of [Figure 3-9: Access Control for User and User Groups](#)

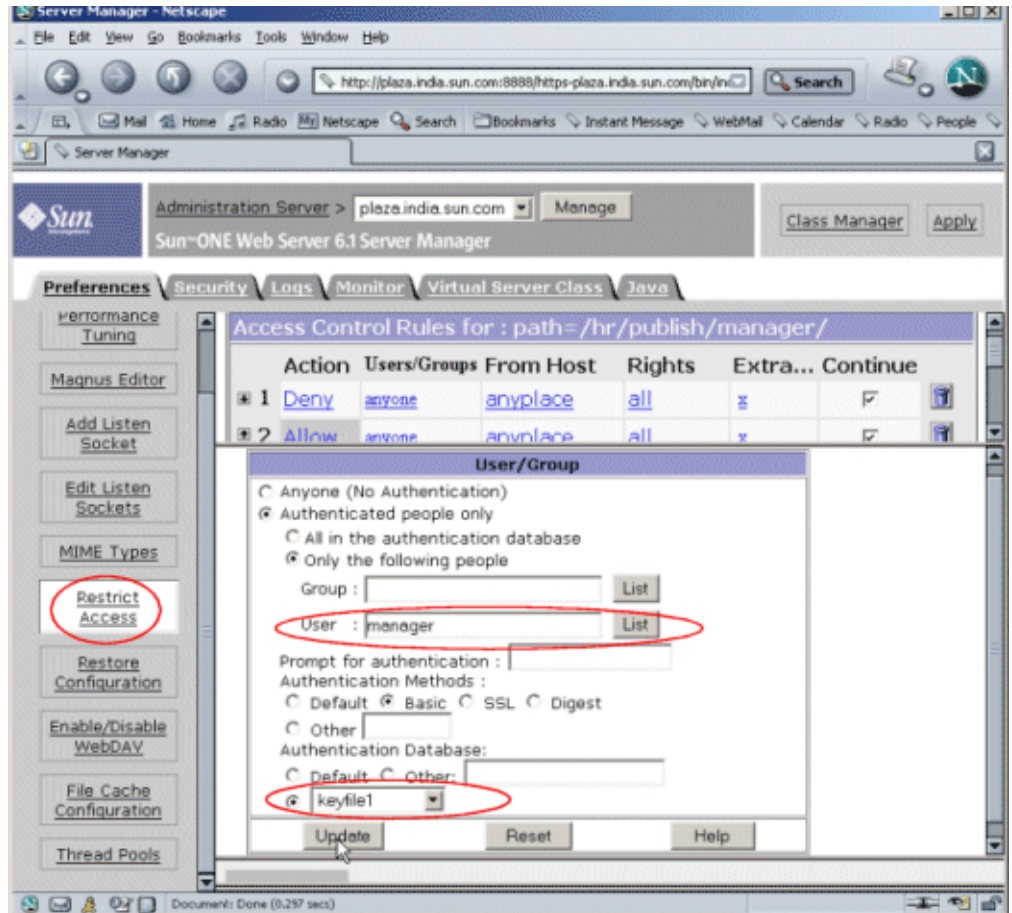


FIGURE 3-9 Access Control for User and User Groups

- 11 Click Update.
- 12 Click Submit to store the new access control rules in the ACL file.
- 13 Click Apply Changes and save and apply the changes that you have made.

Once an ACL is set users accessing the site will be required to authenticate themselves before they are allowed access, as shown below:

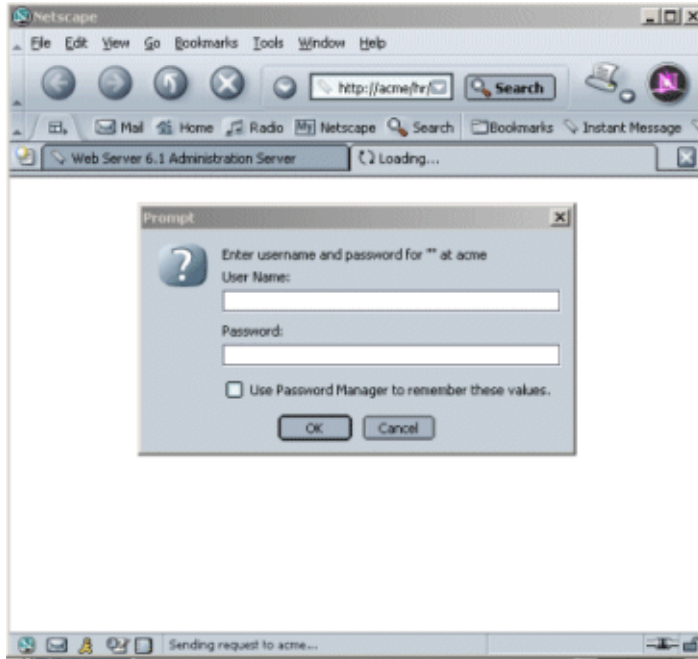


FIGURE 3-10 User Authentication Page

# Publishing Content

---

Sun Java System Web Server 6.1 introduces support for the Web-based Distributed Authoring and Versioning (WebDAV) protocol that enables users to perform remote web content authoring operations. What this means is that once you have created a WebDAV collection on your virtual server users can access remotely.

A complete WebDAV transaction involves a server that supports WebDAV, such as Sun Java System Web Server 6.1, to service requests for WebDAV resources, as well as a WebDAV-enabled client such as Microsoft Internet Explorer, Adobe® GoLive® or Macromedia® DreamWeaver® that supports WebDAV-type web publishing requests.

This chapter includes the following sections:

- [“Set Up the Server to Handle WebDAV” on page 39](#)
- [“Set Up a Client to Use WebDAV” on page 45](#)

## Set Up the Server to Handle WebDAV

Clients can author documents and publish them on your server. You can view, copy, move, and lock documents, and perform other WebDAV operations on collections of resources that are configured to support WebDAV on a virtual server. As server administrator, therefore, you need to create such collections.

For instance, you have a set of documents located in a folder called `/info` in the docroot of your virtual server, `acme`. You want the HR group in Acme Corp. to be able to work remotely with this folder.

### ▼ To enable WebDAV

First you need to enable WebDAV at two levels: globally, at the server instance level and at the virtual server class level. Note that WebDAV is disabled by default.

To enable WebDAV globally:

- 1 Go to the Server Manager and click the Enable/Disable WebDAV link on the Preferences tab.

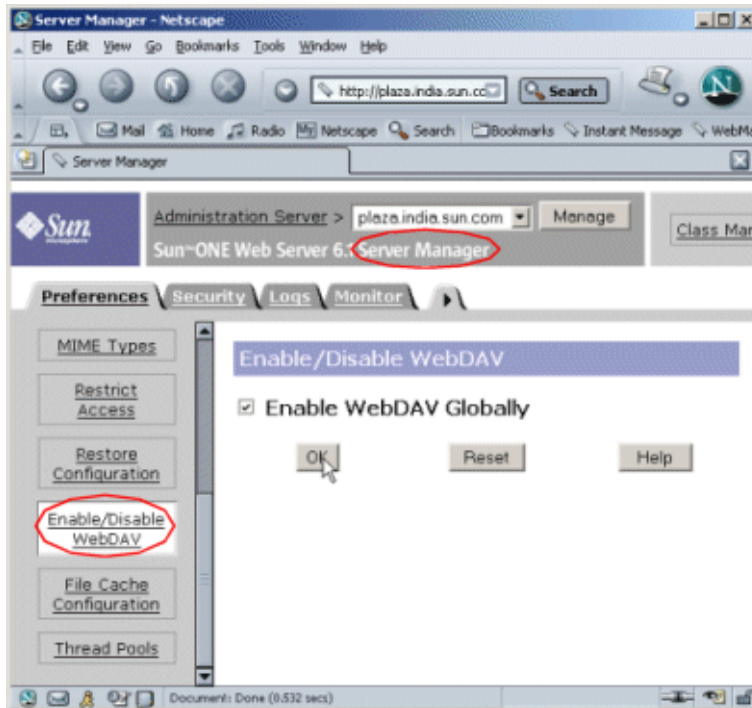


FIGURE 4-1 Enable/Disable WebDAV at Global Level

- 2 Click the Enable WebDAV Globally checkbox.
- 3 Click OK and then click Apply for the changes to take effect.

## ▼ To enable WebDAV for the virtual server

Enable WebDAV for the virtual server `acme`, which belongs to the virtual server class, `vsclass2`:

- 1 Go to the Class Manager interface of `vsclass2`.
- 2 Select the Content Management tab and click the Enable/Disable WebDAV link.



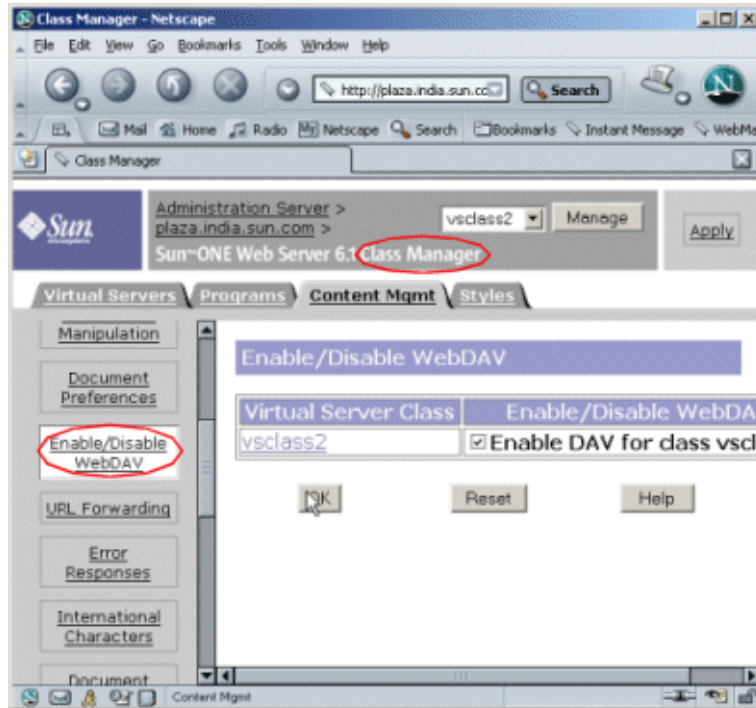


FIGURE 4-2 Enable/Disable WebDAV at Virtual Server Level

3 Check the Enable DAV checkbox corresponding to `vsclass2`.

4 Click OK.

You are now ready to create the WebDAV collection.

At a minimum, creating a collection involves specifying the path to a directory that you want to enable WebDAV on. Now, if this directory contains only static data (such as HTML pages or images), you need to specify only the *URI* path to the directory. However, if the directory also contains dynamic content (such as JSP, SHTML, or CGI programs), you must also specify a *source URI*.

For instance, you have a dynamic page, `test.shtml` in the collection. This page contains a script that displays today's date on the browser. While creating the collection, you specify only a *URI* for the collection, and no *source URI*. In this case, when a client invokes the `test.shtml` page, how it appears on a browser is shown in the [Figure 4-3: Page Displaying the Current Date for the Collection](#):

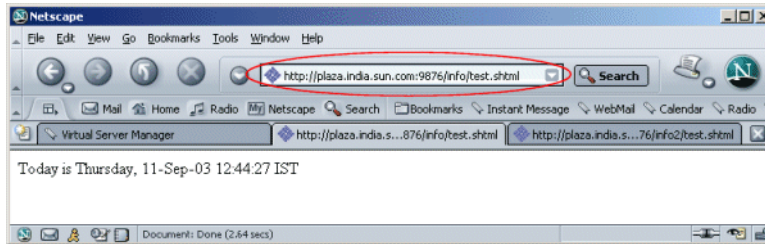


FIGURE 4-3 Page Displaying the Current Date for the Collection

You can also configure a source URI for the collection, say `/info_source`. Now if a client invokes the `test.shtml` page, the figure below shows how it appears on a browser window:

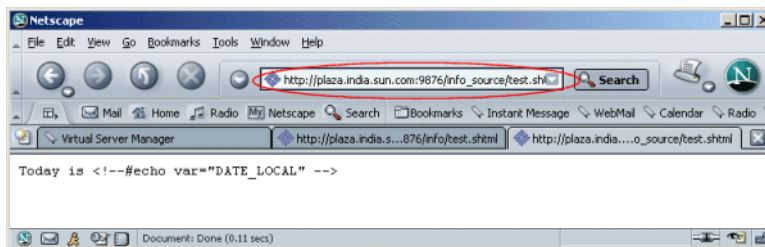


FIGURE 4-4 Page displaying the source URI for the collection

By specifying a source URI for a collection, you expose the source of a resource to clients. This is discussed in [Chapter 3, Enabling Access Control](#).



**Caution** – Exposing the source obviously has security implications and you are recommended to set access control on directories that you map source URIs against.

Create a collection and specify both URI and also a source URI so that the collection can expose the source of dynamic content.

## ▼ To create a WebDAV collection

- 1 Go to the Virtual Server Manager interface page for your virtual server, `acme` and select the WebDAV tab
- 2 Click the Add DAV Collection link
- 3 Specify `/info` as the URI and `/info_source` as the source URI

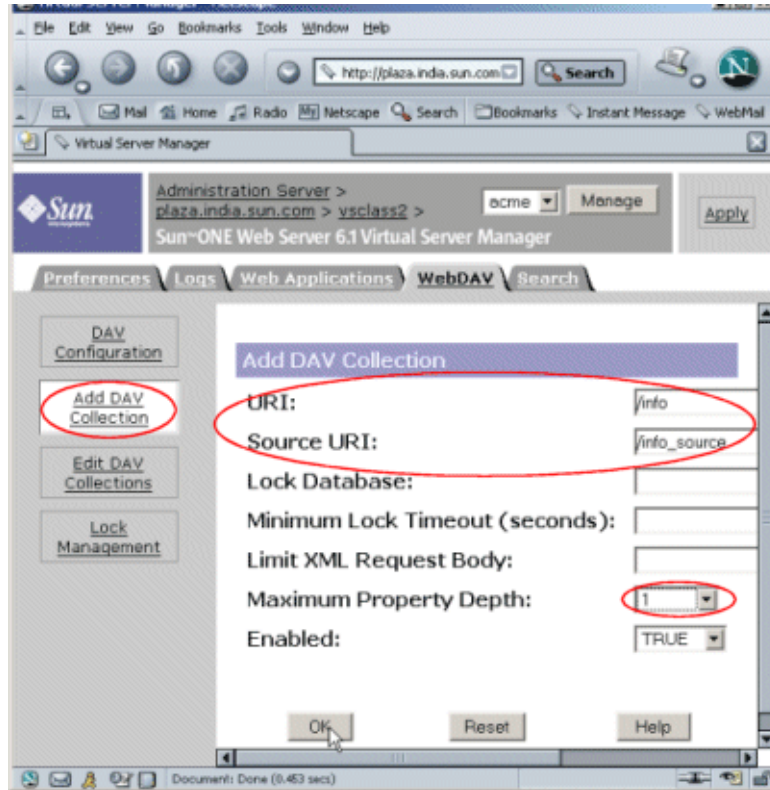


FIGURE 4-5 Adding DAV Collection

- 4 Set the Maximum Property Depth value to either 1 or infinity.
- 5 Click OK and then Apply to save and apply your changes.

You can specify other parameters for the collection in order to fine-tune the level of WebDAV support. For instance, you can set an upper limit on how much data a person can copy onto the server or for what length of time a resource can be locked. You can do this either at the time of creating the collection or at any time after you have created the collection using the Edit DAV Collections page.

Once you are done with creating or editing the collection, you need to ensure that only the user named “manager” and no one else has the right to access the collection.

**Note –**

- You must ensure that filesystem permissions should be owned by the user the server is running as. For example, if `/info` is the URI for a WebDAV collection and if the server is running as a user named “Joe”, ensure that `/info` maps to a directory to which “Joe” has filesystem permissions.
- Because WebDAV gives users the right to modify the content of collections on your server, you must ensure that these collections are protected against unrestricted and unauthorized access. For more information on enabling access control and security, see [Chapter 3, Enabling Access Control](#).

If you need to provide access control and security to the collection that you have just created for only the user named “manager”, use the Edit ACL button on the Edit WebDAV Collections page to set this up.

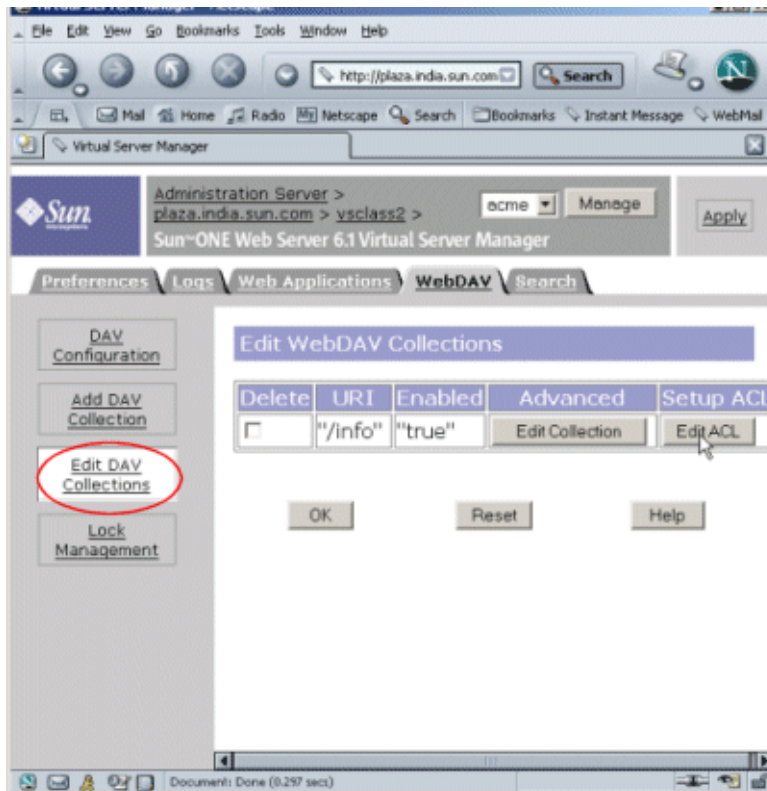


FIGURE 4-6 Editing WebDAV Collection

The server is now set up to handle WebDAV requests.

## Set Up a Client to Use WebDAV

You can use any WebDAV-enabled client software to access and use resources on a WebDAV-enabled server, for example, Microsoft Internet Explorer, Adobe GoLive and Macromedia DreamWeaver.

You would first need to connect to Sun Java System Web Server server using a valid WebDAV URI. For example, if `/info` is the URI to a WebDAV-enabled collection on a server named `acme` running on port 2222, specify the URL for the client to make a connection with the server:

```
http://acme:2222/info
```

On the client-side, you receive a listing of the contents of the collection.

You can perform several operations on the listed resources. For example, creating a collection on the server, download files from the collection to your machine, upload files, lock, unlock, copy, move or delete resources. Typically, if you were making changes to a file, use the WebDAV client software to lock it to prevent others from simultaneously working on the same file and overwriting your changes.

Sun Java System Web Server 6.1 keeps track of the outstanding locks in a locking database.

To view the outstanding locks or to free a locked resource, you use the Lock Management link on the Virtual Server Manager interface, as shown in the [Figure 4-7: Lock Management Link](#) :

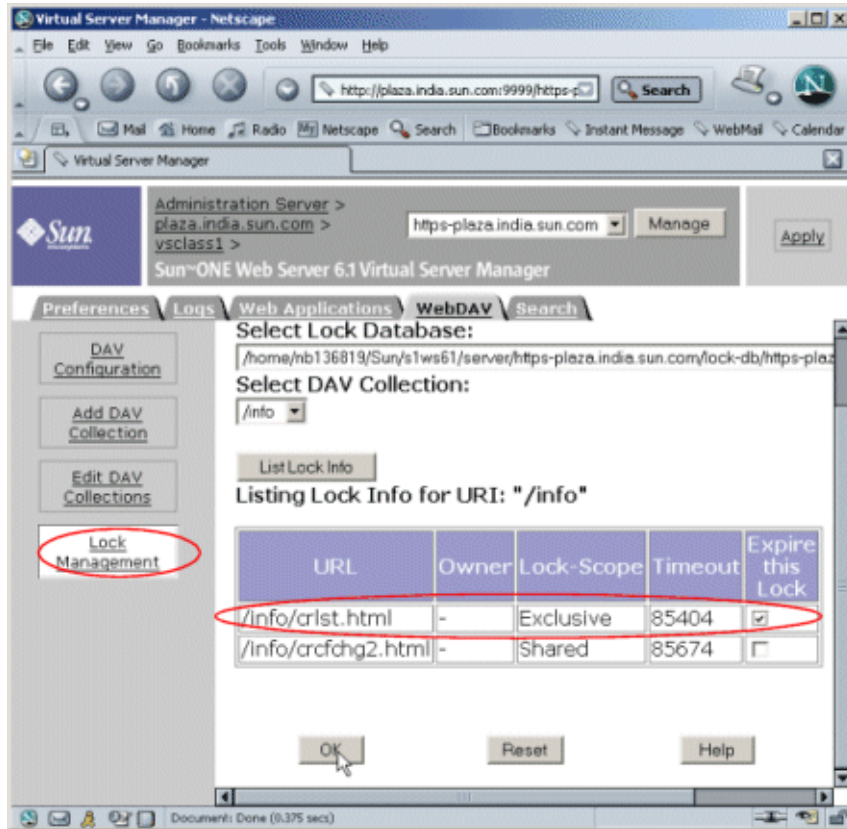


FIGURE 4-7 Lock Management Link

For more information on WebDAV functionality, see the *Sun Java System Web Server 6.1 SP7 Administrator's Guide*.

## Making Content Searchable

---

This chapter describes how you can search content on a virtual server. Sun Java System Web Server 6.1 includes a search engine that allows you to search documents on the server and display results on a web page. It provides a default search web application which you can enable or disable on a specific virtual server. This search application provides the basic web pages used to query collections and view results.

Search can be enabled on a virtual server, ensure that Java is enabled for the virtual server class that contains the virtual server. Java is enabled by default and you can enable/disable it per server instance in Sun Java System Web Server 6.1 The Java tab for the virtual server contains a link, the Enable/Disable Java link, that takes you to a page which allows you to enable or disable Java.

Next, because searches require a database of searchable documents (called a collection) against which users can search, you need to create this database. At a minimum, you must have at least one collection configured on your server. You can however create as many collections as required.

This chapter includes the following sections:

- [“Create a Search Collection” on page 47](#)
- [“Update Your Collection” on page 49](#)
- [“Enable the Search Application” on page 50](#)
- [“Customize the Search Application” on page 51](#)

### Create a Search Collection

Let’s say you want to create a collection called “HR” on your virtual server acme. To do so, use the Create Collection page in the Search tab of your virtual server.



FIGURE 5-1 Creating a Search collection

In the figure above, certain examples are specified. These parameters define our collection.

- `/hr/publish/info` is the directory from which documents will be indexed by the collection. By default, only the directory designated as the document root directory for the virtual server is listed for selection. To add a directory to the list of indexable directories, you must create additional document directories, as discussed in [Chapter 2, Creating and Using Virtual Servers](#)
- “HR” is the collection name. The server uses this internally to refer to the collection.
- “HR Docs” is the display name. This is what end users will see as the name of the collection in the search query page. If this field is left blank, the collection name “HR”, would appear on the search query page.
- `*.html` as the value of the Pattern field indicates that only files with a `.html` extension will be indexed.
- The Encoding field is left as is. This means that the default encoding, “ISO-8859-1” will be used. If, for instance, the documents you want to index are in Japanese, you might use an appropriate encoding such as, “Shift\_JIS.”



You need to save and apply your changes after updating the page for the changes to take effect.

You can update any of the parameters in the search collections you have created at any time using the Update Collection link on the Search tab of the virtual server.

## Update Your Collection

When you add new documents to the /hr/publish/info folder, update the “HR” collection to index the new documents into the collection by manually using the Update Collections page.

If your site is constantly being updated, schedule regular maintenance tasks on the collection. You might for example, wish to update your collections at the end of the day when you know that new documents are likely to have been added.

To automate maintenance tasks on collections, use the Add Scheduled Maintenance link in the Search tab of the virtual server. Choose Update as the type of task you want to automate, and specify other parameters as appropriate, for example, the time of day to perform the task, the day(s) of the week to perform the task, figure as shown below:

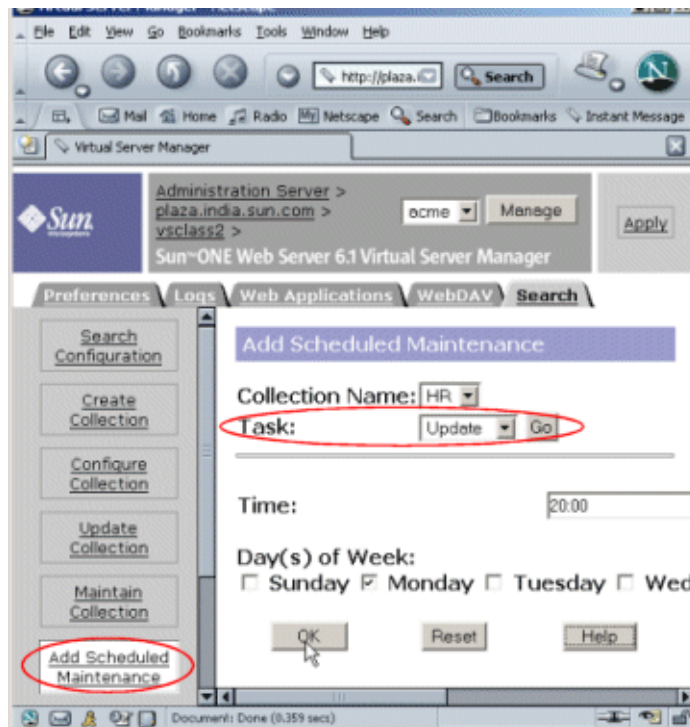


FIGURE 5–2 Add Scheduled Maintenance Link

## Enable the Search Application

Next, we need to enable the search application on the virtual server using the Search Configuration link in the Search tab of the virtual server.

If you plan to use the default search application, click the Enabled checkbox on this page and use the default values for the rest of the parameters.

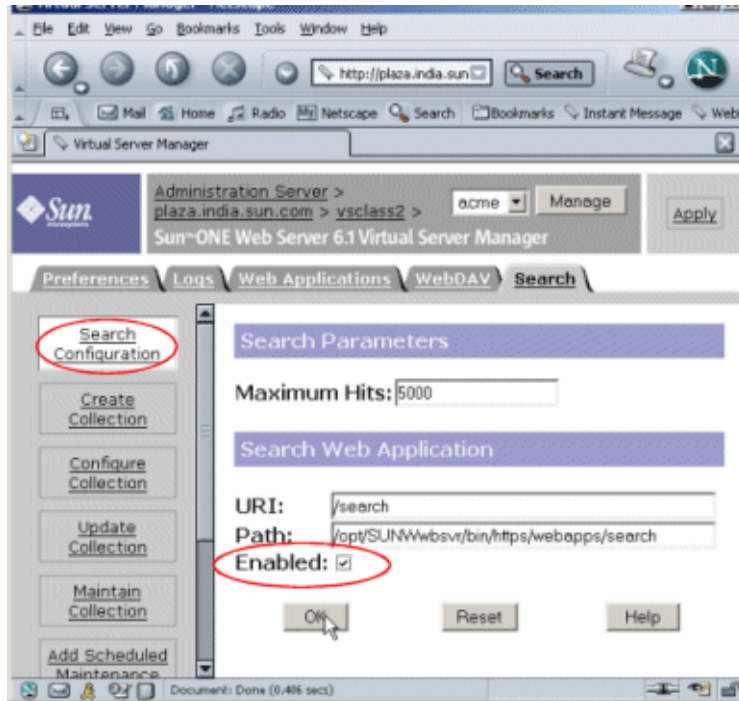


FIGURE 5-3 Enabling the Search Application

To view the search query page, enter the following address on a browser window:

```
http://acme/search
```

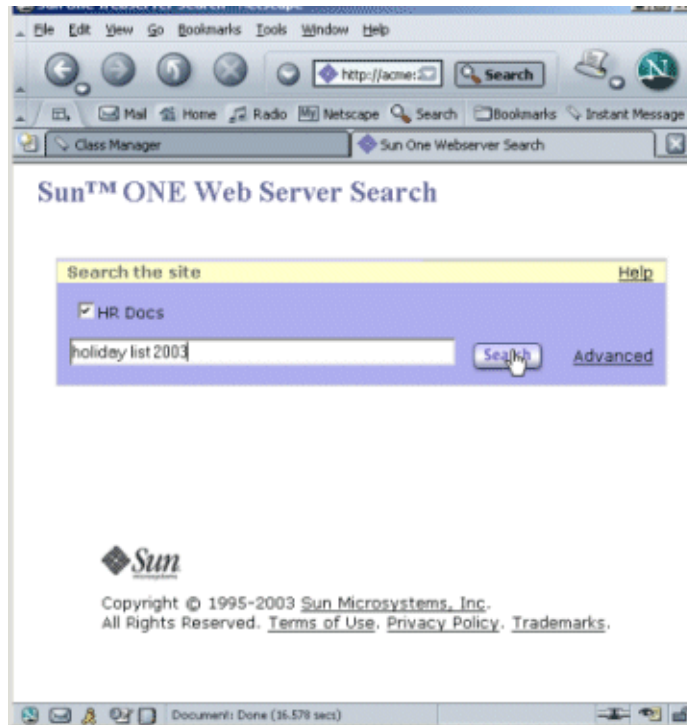


FIGURE 5-4 Search Query Page

## Customize the Search Application

You may choose to customize the search application to suit your specific needs. You can use the JSP tag libraries included with Sun Java System Web Server 6.1 to do so. Take a look at the default search application located at `server_root/bin/https/webapps/search` as a sample application that illustrates the use of customizable search tags. The section on Customizing Search Pages in the Sun Java System Web Server 6.1 *Administrator's Guide* provides examples of customizing the search page, by having the query form appear on a side bar of the page or search results appear differently.

For a detailed description of the JSP search tags you can use, see the *Sun Java System Web Server 6.1 SP7 Programmer's Guide to Web Applications*.



# Running Java Web Applications On Your Server

---

Sun Java System Web Server 6.1 allows you to create and deploy web applications that extend your server dynamically. Sun Java System Web Server 6.1 provides a runtime environment, the web container, that supports web applications. The web container supports the Java™ Servlet 2.3 API specification and the JavaServer Pages™ (JSP™) 1.2 specification, which allow servlets and JSPs to be included in web applications.

## Deploying Web Application

A web application is a collection of servlets, JSPs, HTML documents, and other web resources which might include image files, compressed archives, and other data.

To deploy a web application using the web-based Sun Java System Web Server interface, you need to specify where the web archive (WAR) file for the application is located.

In this chapter, we will use a sample application that is bundled with Sun Java System Web Server to walk through the steps involved in deployment. For details on how to create a web application and write servlets and JSPs, see the *Sun Java System Web Server 6.1 SP7 Programmer's Guide to Web Applications*.

The WAR file is located in the `server_root/plugins/java/samples/webapps/simple` directory, `webapps-simple.war`, for a simple web application.

### ▼ To deploy web application

- 1 Access the virtual server and click on the Web Applications tab

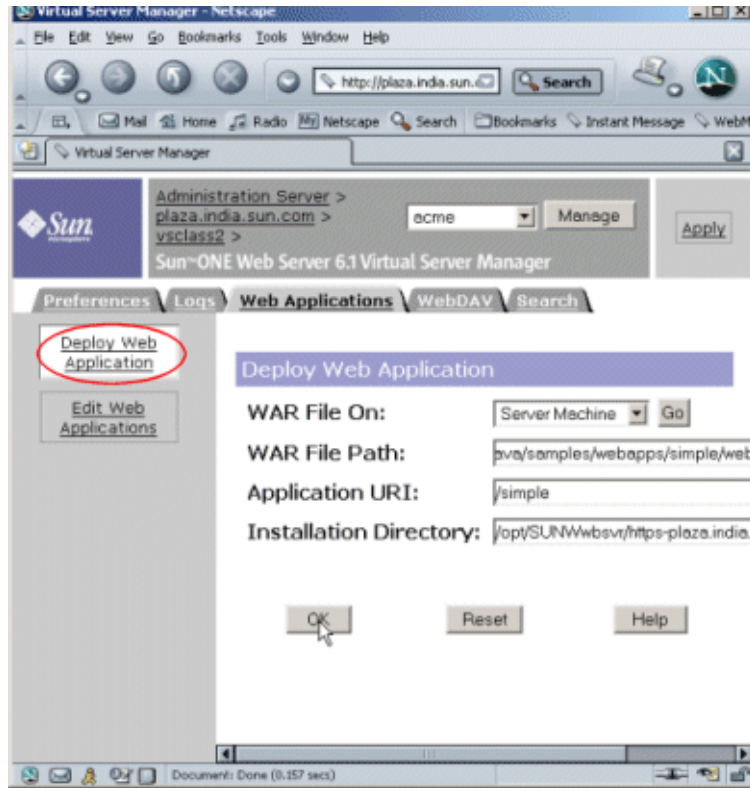


FIGURE 6-1 Deploying Web Application

- 2 Select **Server Machine** from the “WAR File On” drop-down list and click the **Go** button.
- 3 Specify the following:
  - The absolute path to the sample WAR file, `webapps-simple.war`:  
`<install_root>/plugins/java/samples/webapps/simple/webapps-simple.war`
  - The application URI:  
`/simple`.  
 You do not make any changes to the default installation directory
- 4 Click **OK** and then **Apply** to save and apply your changes.
- 5 To launch the application, type the following URL in a browser window:  
`http://server-name:port/simple`

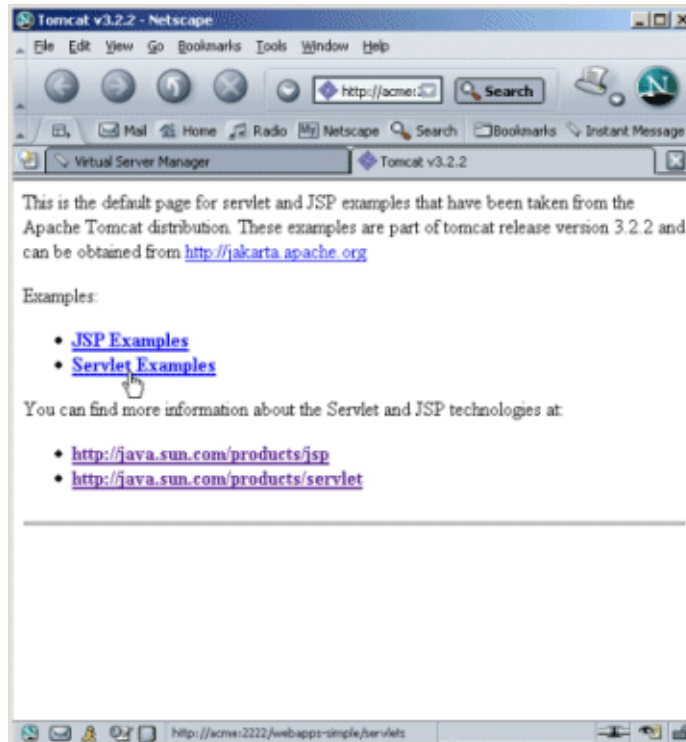


FIGURE 6-2 Web Application Page — Sample 1

The web application contains a listing of several sample web applications that you can run. Traverse the JSP and servlet links to execute each sample web application, as shown in [Figure 6-3: Web Application Page - Sample 2](#):

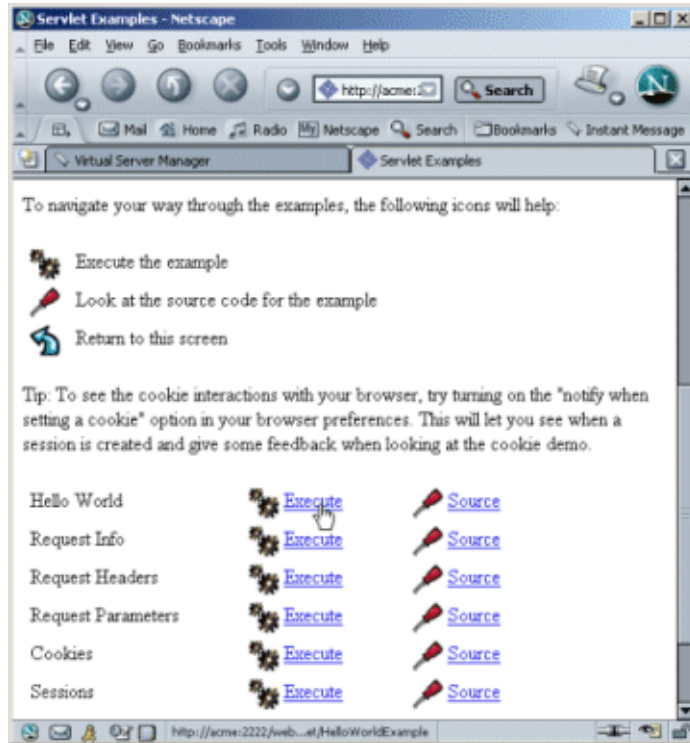


FIGURE 6-3 Web Application Page — Sample 2

The page `server_root/plugins/java/samples/index.html` is a good place to begin exploring and running the sample applications that are bundled with Sun Java System Web Server. For more information about how to set up database connectivity and use Java resources in web applications, see [Chapter 7, Setting Up Database Connectivity](#).



# Setting Up Database Connectivity

---

Web applications leverage the benefits of the component-based Java™ 2 Platform, Standard Edition (J2SE) technology, which include scalability and speed.

Web components, Java Servlets or JSP pages provide the dynamic extension capabilities for Sun Java System Web Server 6.1. Servlets and JSP pages are supported by the services of a runtime platform called a web container. The web container provides services such as request dispatching, security, and concurrency. It also gives web components access to APIs such as naming, datasources, and e-mail.

Sample application that uses the Java Database Connectivity (JDBC) API to record greetings to a database is illustrated in this chapter.

## JDBC Settings

In this section you can define the necessary JDBC-related settings in the web server environment prior to deploying and exercising the sample.

1. [“Install and Configure the JDBC driver” on page 57](#)
2. [“To start the database server” on page 58](#)
3. [“Create a Database and Table” on page 59](#)
4. [“To define the JDBC connection pool” on page 59](#)
5. [“To define the JDBC resource” on page 62](#)

## Install and Configure the JDBC driver

JDBC configuration involves setting up a JDBC driver, defining a JDBC connection pool, and registering the JDBC resources used by your application.

A JDBC driver must be configured in the classpath of the web server before you can exercise applications that use JDBC. In this example, the PointBase JDBC driver is used.

To install and configure PointBase for use with the sample applications, follow these steps:

1. Download and Install PointBase Server and Client Products
2. To add PointBase Type 4 JDBC Driver to Classpath of the Web Server

## Download and Install PointBase Server and Client Products

Download the PointBase evaluation software from:

<http://www.pointbase.com/>

Install the PointBase Server and Client products on your system. The PointBase Client product includes the PointBase Type 4 JDBC driver.

### ▼ To add PointBase Type 4 JDBC Driver to Classpath of the Web Server

- 1 **Open the *server-instance/config/server.xml* file of the web server instance. Add the location of the PointBase driver in the *classpathsuffix*.**

You can find the JDBC driver under `<pointbase_install_dir>/lib/`.

The driver is named `pbclientnn.jar` where *nn* represents the version of PointBase.

- 2 **Restart the web server to make the server aware of the driver.**

Alternatively, you can specify the location of the PointBase driver in the Classpath Suffix field in configuration of the Web Server:

- a. **Check Java tab of the Web Server instance -> Java Path Settings page to make this change.**

If a PointBase JDBC driver is already configured in the Classpath Suffix field, replace it with the path to the newly installed driver.

- b. **Once you have changed the Classpath Suffix field, apply the changes and restart the web server instance.**

### ▼ To start the database server

- ▶ **The database server can be started by performing one of the following actions:**

- **On Windows Platforms:**

Execute: `<pointbase_install_dir>\tools\server\startserver.bat`

- **On UNIX Platforms:**

Execute: `<pointbase_install_dir>/tools/server/startserver.sh`

Once you execute the startup script, the following text will be displayed:

```
Server started, listening on port 9092, display level: 0 ...
```

---

## Create a Database and Table

For information about creating and maintaining a database using PointBase, refer to the database documentation. After you have created a database, you need to use the following script to create the table 'Greeting', which will record application entries:

```
<install_root>/plugins/java/samples/webapps/jdbc/simple/src/sql/jdbc-simple-pb.sql
```

### ▼ To define the JDBC connection pool

Before running the sample application, you need to define a suitable JDBC connection pool that maps to the PointBase database server. You also need a JDBC resource that associates the JDBC references made in the sample application to the JDBC connection pool definition.

- 1 Through a web browser, access the administration server interface.

---

**Note** – You might have to start the administration server if it is not running.

---

- 2 Select the Java tab in the server instance and click on JDBC Connection Pools

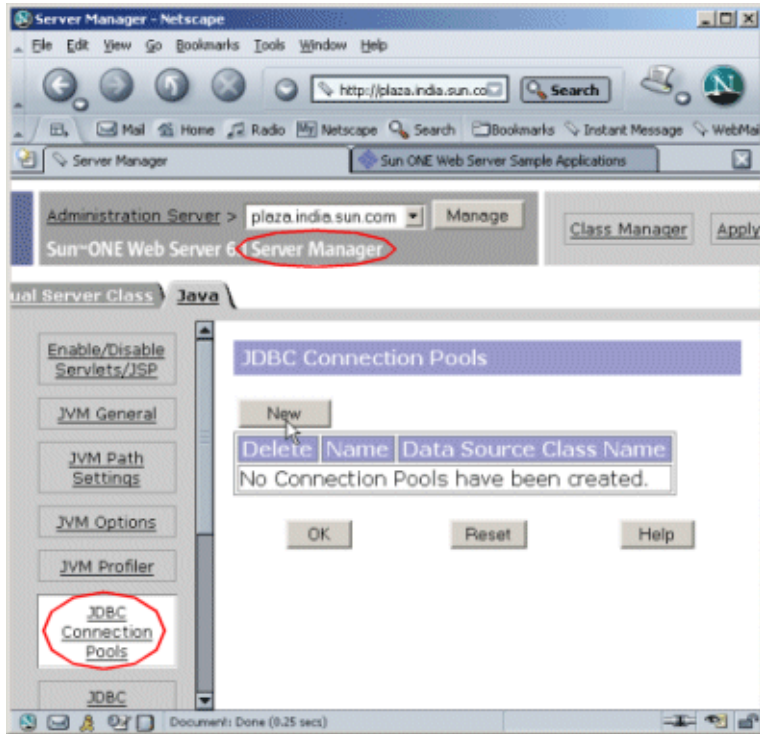


FIGURE 7-1 Creating New JDBC Connection Pool

- 3 Click New, and select the JDBC driver, as shown in [Figure 7-2: Selecting the JDBC Driver:](#)

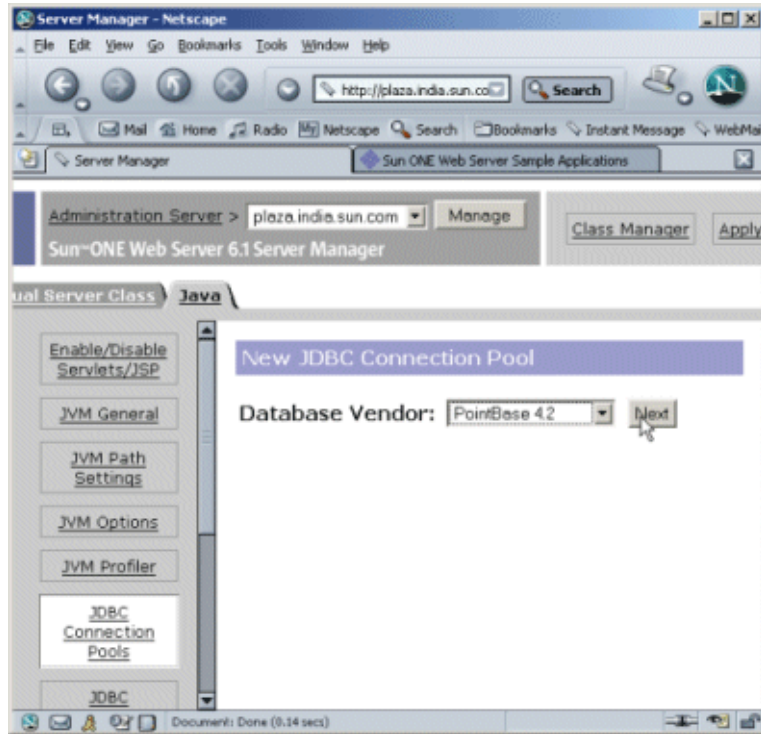


FIGURE 7-2 Selecting the JDBC Driver

- 4 In the Add New JDBC Connection Pools page, specify the following settings and the rest unmodified:
  - jdbc-simple-pool as the name of the new connection pool.
    - jdbc:pointbase:server://localhost/sample as database name
    - PBPUBLIC as the user name
    - PBPUBLIC as the password
    - Transaction Isolation set to read-committed

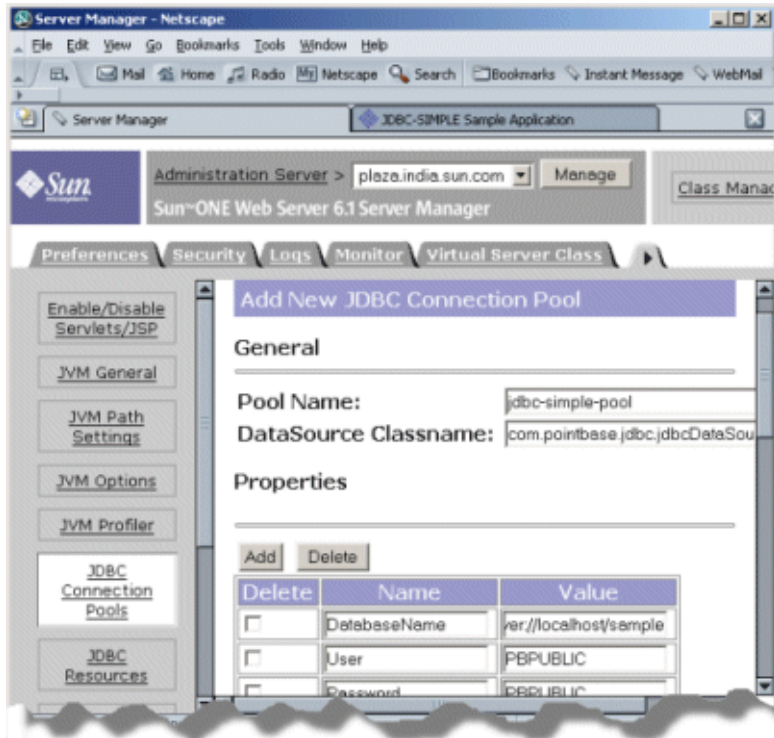


FIGURE 7-3 JDBC Connection Pool — General Information

- 5 Click OK and then Apply to save and apply your changes.

## ▼ To define the JDBC resource

Now that the JDBC connection pool definition has been created, you are ready to define a JDBC resource and associate it with the connection pool entry.

- 1 Under the Java tab of the server instance, click the JDBC Resources link.
- 2 Click the New button to define a new resource entry.

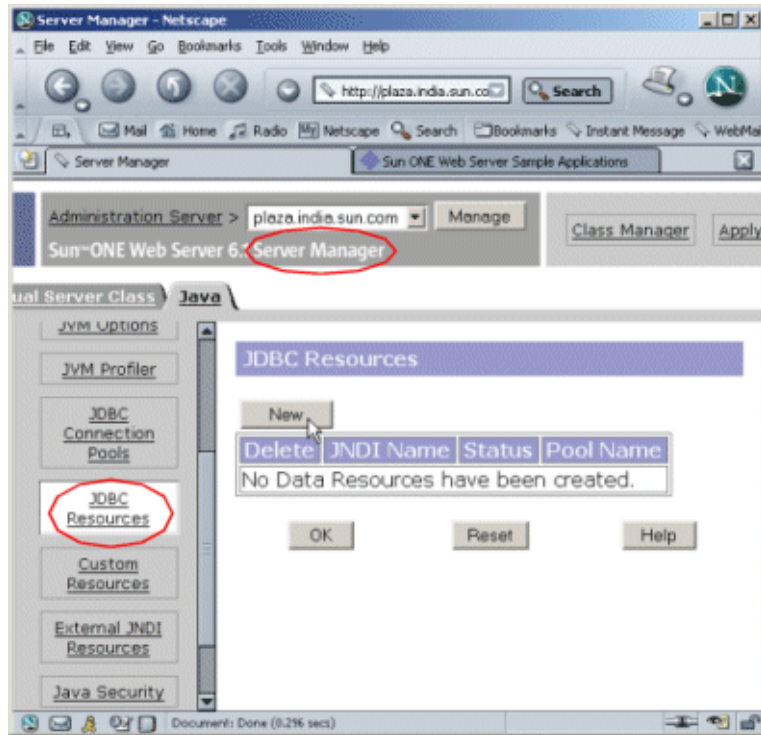


FIGURE 7-4 Creating a JDBC Resource

- 3 Since the `sun-web.xml` file of the sample web application refers to the JNDI Name of the resource as "jdbc/jdbc-simple", you need to define the JDBC resource with this same JNDI name.

Set the fields to the following values and click on the OK button to define the resource:

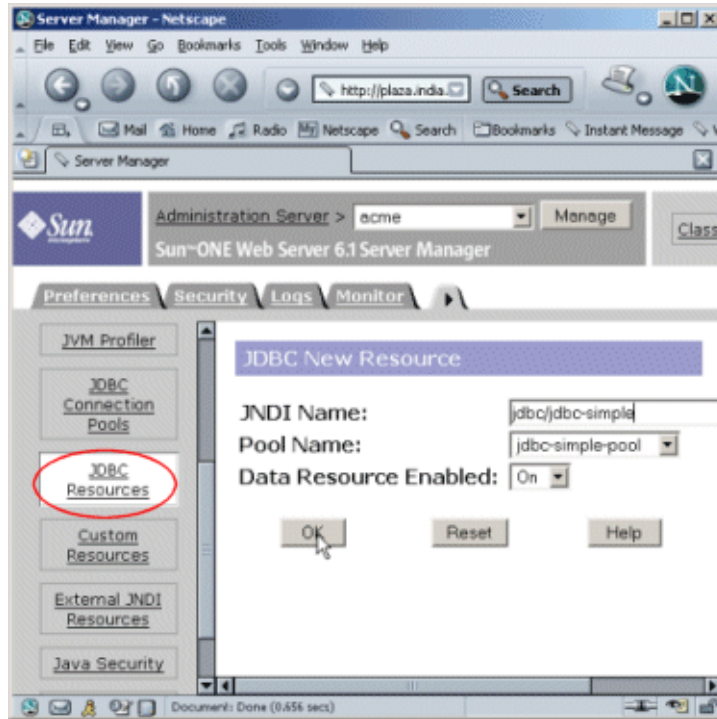


FIGURE 7-5 JDBC Resource — General Information

**4 Click on OK and Apply to save and apply your changes.**

After you've set up database connectivity, you are ready to deploy the `jdbc-simple` sample web application that uses these newly created JDBC resources. You can find this sample application at the following location:

`install_root/plugins/java/samples/webapps/jdbc/simple/jdbc-simple.war`

To deploy this sample application, follow the steps outlined in [Chapter 6, Running Java Web Applications On Your Server](#).



## Using CGI

---

You can set up your virtual server to serve dynamic content in two ways: using Java-based web applications and using non-Java applications such as CGIs, ASP, Server-side HTML (SHTML) tags and NSAPI programs. This chapter describes how to use CGI programs. In the following chapter, [Chapter 9, Using Sun Java System Active Server Pages](#) we will use Sun™ Java System Active Server pages to create dynamic content.

For details about using SHTML tags, see the *Sun Java System Web Server 6.1 SP7 Programmer's Guide*, and for details on using NSAPI, see the *Sun Java System Web Server 6.1 SP7 NSAPI Programmer's Guide*.

CGI (Common Gateway Interface) programs are external programs that the server calls to process data.

Sun Java System Web Server 6.1 bundles a Perl interpreter, which makes it possible to run CGI programs that are written in the Perl programming language. The Perl interpreter is located at `server_root/bin/https/perl/perl`. On Windows, file associations need to be in place for CGI programs to run. For example, files with a `.pl` filename extension need to be associated with the Perl interpreter executable before they can be executed as CGI programs.

For information about writing CGI programs, see the following sources of information:

- *Sun Java System Web Server 6.1 SP7 Programmer's Guide*
- The Common Gateway Interface at:  
<http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>

## Configuring Server to use CGI

This section uses a simple exercise to show how you can configure your server to use CGI.

For instance, you have a simple Perl program called `hello.pl` that we want to use CGI on our web server to execute. When executed, this program will display the words “Hello World!” on a browser window.

### ▼ To configure your server to use CGI

- 1 **Use a text editor to create a file called `hello.pl` and copy the following lines into it (remember to substitute `<server_root>` with the absolute path to your server root directory), and save your changes:**

```
#!<server_root>/bin/https/perl/perl
print "Content-type:text/html\n\n";
print "Hello World!";
```

- 2 **Create a directory for this program. This directory is known as the `cgi-bin` directory, you can rename it if required.**

```
mkdir /opt/SUNWwbsvr/cgi-bin
```

- 3 **Copy the `hello.pl` file into the `/opt/SUNWwbsvr/cgi-bin` directory**
- 4 **Access the Server Manager interface on a browser and click on the Programs tab.**

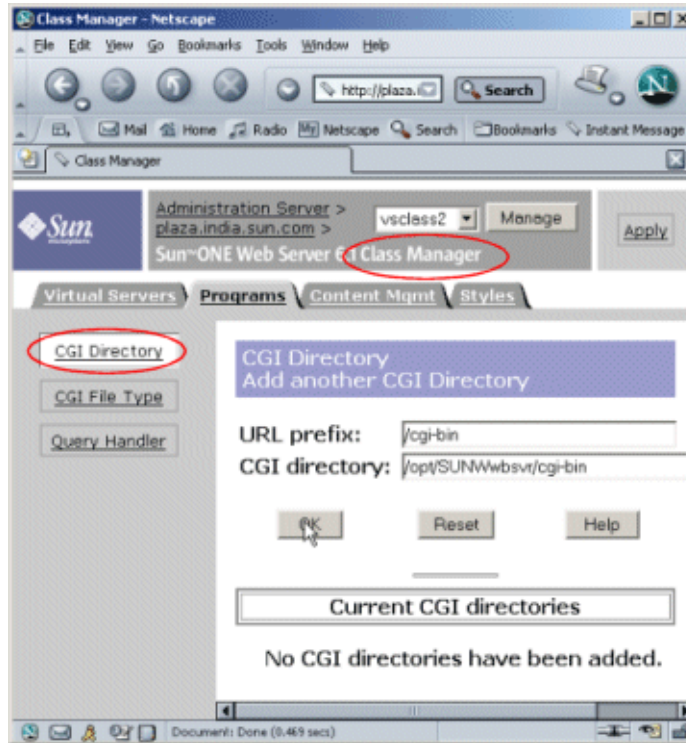


FIGURE 8-1 Adding a CGI Directory

- 5 Click the **CGI Directory** link if the server is running on UNIX. For Windows, the **WinCGI Directory** link. Specify:
  - a URL prefix (to which clients will send requests). In this example, this is set to `/cgi-bin`
  - the CGI directory. In our example, `/opt/SUNWwbsvr/cgi-bin`.




---

**Caution** – The server must have read and execute permissions to this directory.

---

- 6 Click **OK** and then **Apply** to save and apply the changes.

Open a browser window and (if your server is called `acme`, running on port 2222), type `http://acme:2222/cgi-bin/hello.pl`.

This executes the “Hello World” script, as shown in [Figure 8-2: Sample CGI Script](#):

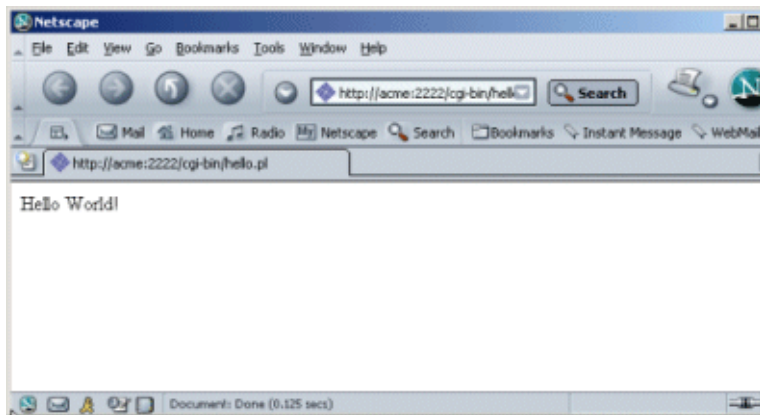


FIGURE 8-2 Sample CGI Script

## Using Sun Java System Active Server Pages

---

Sun Java System Web Server 6.1 supports Sun™ Java System Active Server Pages version 4.0.1 (formerly Sun Chili!Soft ASP). This means that the Web server passes processing of all pages having the .asp file name extension to the Sun Java System Active Server Pages Server.

To use the Active Server Pages Server, you need to install and configure Sun Java System Active Server Pages against the Web server. The Sun Java System Active Server Pages installer is available on the Companion CD if you purchased the Sun Java System Web Server Media Kit, or by download from the following location:

<http://www.sun.com/software/chilisoft/index.xml>

You need to log in as root to install Sun Java System Active Server Pages.

### Administering ASPs

When you install Sun Java System Active Server Pages, you can choose to configure it against any of the Web server instances that the Active Server Pages installer automatically detects. The Active Server Pages installer configures a URL to connect from a browser. For example;  
`http://plaza:5100`

To administer Sun Java System Active Server Pages, type the specified URL in a browser window.



FIGURE 9-1 Administering ASP

You can fine-tune the Active Server Pages server to work with the Web server. See the product documentation included with Sun Java System Active Server Pages for more details on what you need to do to achieve this.

To deploy a simple application, `index.asp`, which displays the current date, and run it off the Web Server.

## ▼ To deploy a simple application

- 1 Use a text editor to create a file called `index.asp` and copy the following lines into it:

```
<html>
<body>
<h1> Today's date is :<%response.write (date())%> </h1>
</body>
</html>
```

- 2 **Create a directory for this program.**  

```
mkdir /opt/casp/sampleapp
```
- 3 **Copy the `index.asp` file into the newly created `/opt/casp/sampleapp` directory.**
- 4 **Access the Active Server Pages Server interface on a browser and click on the “Add a new application” link.**



FIGURE 9-2 ASP Console — Adding a New application

- 5 **In the Add Application page, specify `sample` as the name of the application and `/opt/casp/sampleapp` as the directory name, and then click Save.**

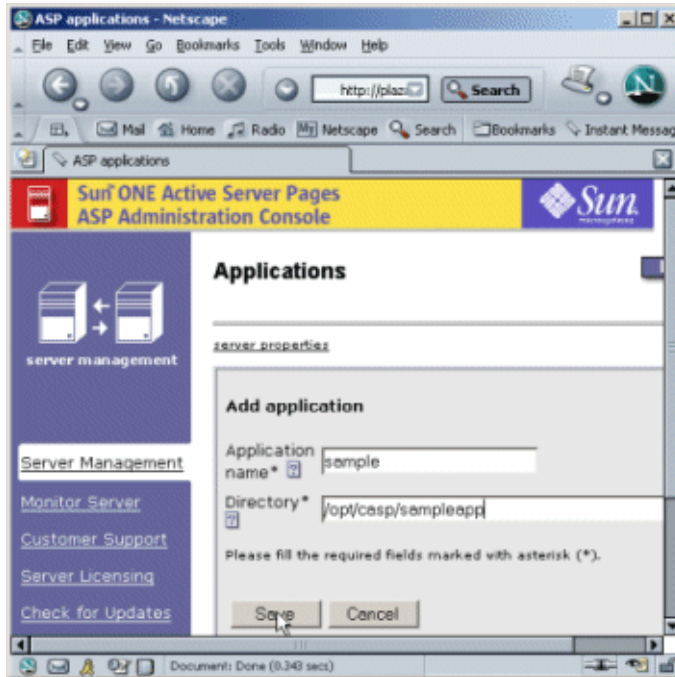


FIGURE 9-3 ASP Application — General Information

**6 Open a browser window and type the following URL:**

`http://web-server-hostname/sample/index.asp`

The sample application displays the current date in the browser.



FIGURE 9-4 Sample ASP



# Index

---

## A

- Access the Administration Server, 13-14
- Access the Web Server, 15
- Add a User, 30-31
- Add New JDBC Connection Pools page, 61
- Additional document directories, 15
- Administration interface, more information about, 6
- Administration Server, 13
- Administration Server Instance, 11
- Allow /Deny page, 35
- ASP, 65
  - Downloading Sun Java System ASP, 69
  - Using Sun Java System Active Server Pages, 69-72

## C

- CGI
  - cgi-bin directory, 66
  - Perl interpreter, 65
  - Using CGI, 65-67
- cgi-bin directory, 66
- Change the Document Root of a Virtual Server, 25-26
- Class Manager, 13, 22
- Classpath, 58
- Create a Directory Service, 28-30
- Create a Listen Socket, 19-20
- Create a Search Collection, 47-49
- Create a Virtual Server, 21-23
- Create a Virtual Server Class, 20-21
- Create a WebDAV Collection, 42-44
- Create an Additional Document Directory, 23-24

## D

- Default search web application, 47
- Define the JDBC Resource, 62-64
- Deploying Java Web Applications On Your Server, 53-56
- directory service, 28
- document root, 11
- Documentation roadmap, 5-7

## E

- Enabling Access Control, 27-37
- Enabling WebDAV, at virtual server class level, 40

## H

- hardware virtual servers, 18
- Host-based access control, 27
- HTTP Host Header, 18

## I

- Install and Configure a JDBC driver, 57-58

## J

- J2SE, 57
- Java Database Connectivity (JDBC), 57
- Java Enterprise System (JES), 6

Java security constraints, 27  
JDBC, 57  
JDBC Connection Pool, 59-62  
JES, 6  
JNDI Name, 63  
JSP tag libraries, 51

## **K**

Key File, 28

## **L**

listen socket, 18

## **M**

Making Content Searchable, 47-51

## **N**

native access control, 27  
NSAPI, 65

## **P**

Perl interpreter, 65  
PointBase Server, 58  
Publish additional content, 15

## **S**

sample web application, 64  
Search  
    Add Scheduled Maintenance, 49  
    collection, 47  
    collection name, 48  
    create a collection, 47-49  
    customizable search tags, 51

## Search (*Continued*)

    customize the Search Application, 51  
    display name, 48  
    Encoding, 48  
    JSP tag libraries, 51  
    JSP tags, 51  
    Pattern, 48  
    viewing the search query page, 50  
Server Manager, 13  
Server-side HTML, 65  
Set Up a Client to Use WebDAV, 45-46  
Setting Up Database Connectivity, 57-64  
software virtual server., 18  
Source URI, 41  
Specify a Directory Service for a Virtual Server, 31-33  
Start the Administration Server, 12-13  
Start the Web Server Instance, 14  
Sun Chili!Soft ASP, 69  
Sun Java Enterprise System (JES), 6

## **U**

URI, 41  
URL, 11  
User-based access control, 27  
User/Group page, 36  
Using ASP, 69-72  
Using CGI, 65-67  
Using Virtual Servers, 17-26

## **V**

Virtual server, 17  
virtual server class, 18  
Virtual Server Manager, 13, 22

## **W**

Web Applications  
    running Java Web Applications On Your  
    Server, 53-56  
    sample applications, 64

- 
- Web Applications (*Continued*)
    - web container, 53
  - web archive (WAR) file, 53
  - Web-based Distributed Authoring and Versioning (WebDAV) protocol, 39
  - web container, 53, 57
  - Web Server Instance, 11
  - WebDAV
    - access control, restricting access, 44
    - Create collection, 42-44
    - Default search web application, 47
    - Edit ACL, 44
    - Enabling WebDAV globally, 39
    - filesystem permissions, 44
    - locking database, 45
    - Set Up a Client to Use WebDAV, 45-46
    - Set Up the Server to Handle WebDAV, 39-44
    - Source URI, 41
    - URI, 41
  - What are CGI programs?, 65
  - What is a web application?, 53
  - What is access control?, 27

