

Getting Started Guide

Sun™ ONE Web Server

Version 6.1

817-1829-10
August 2003

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

Copyright 2003 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Sun ONE, iPlanet, and all Sun, Java, and Sun ONE based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Netscape is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries.

Adobe GoLive is a registered trademark of Adobe Systems Incorporated in the United States and other countries.

Macromedia DreamWeaver is a registered trademark of Macromedia, Inc. in the United States and other countries.

Pointbase is a registered trademark of Pointbase, Inc. in the United States and other countries.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of Sun Microsystems, Inc. and its licensors, if any.

THIS 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 2003 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, Sun ONE, et iPlanet sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d'autres pays.

UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Netscape est une marque de Netscape Communications Corporation aux Etats-Unis et dans d'autres pays.

Adobe GoLive est une marque enregistrée de Adobe Systems Incorporated, Inc aux Etats-Unis et dans d'autres pays.

Macromedia DreamWeaver est une marque enregistrée de Macromedia, Inc aux Etats-Unis et dans d'autres pays.

Pointbase est une marque enregistrée de Pointbase, Inc. aux Etats-Unis et dans d'autres pays.

Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite préalable de Sun Microsystems, Inc. et, le cas échéant, de ses bailleurs de licence.

CETTE DOCUMENTATION EST FOURNIE "EN L'ÉTAT", ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

About This Guide	5
Chapter 1 Basic Tasks	11
Chapter 2 Creating and Using Virtual Servers	17
Chapter 3 Enabling Access Control	27
Chapter 4 Publishing Content	39
Chapter 5 Making Content Searchable	49
Chapter 6 Running Java Web Applications On Your Server	55
Chapter 7 Setting Up Database Connectivity	59
Chapter 8 Using CGI	67
Chapter 9 Using Sun ONE Active Server Pages	71
Index	75

About This Guide

This Getting Started Guide is recommended for first time users of the Sun ONE Web Server 6.1 product. Through hands-on exercises, this guide introduces you to the basic features of the server.

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

This preface contains information about the following topics:

- [Using the Documentation](#)
- [How This Guide Is Organized](#)
- [Documentation Conventions](#)
- [Product Support](#)

Using the Documentation

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

<http://docs.sun.com/prod/sunone>

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

Table 1 Sun ONE Web Server Documentation Roadmap

For Information About	See the Following
Late-breaking information about the software and documentation	<i>Release Notes</i>

Table 1 Sun ONE Web Server Documentation Roadmap

For Information About	See the Following
Getting started with Sun ONE Web Server, including hands-on exercises that introduce server basics and features (recommended for first-time users)	<i>Getting Started Guide</i>
Performing installation and migration tasks: <ul style="list-style-type: none">• Installing Sun ONE Web Server and its various components, supported platforms, and environments• Migrating from Sun ONE Web Server 4.1 or 6.0 to Sun ONE Web Server 6.1	<i>Installation and Migration Guide</i>

Table 1 Sun ONE Web Server Documentation Roadmap

For Information About	See the Following
Performing the following administration tasks: <ul style="list-style-type: none">• Using the Administration and command-line interfaces• Configuring server preferences• Using server instances• Monitoring and logging server activity• Using certificates and public key cryptography to secure the server• Configuring access control to secure the server• Using Java™ 2 Platform, Enterprise Edition (J2EE™ platform) security features• Deploying applications• Managing virtual servers• Defining server workload and sizing the system to meet performance needs• Searching the contents and attributes of server documents, and creating a text search interface• Configuring the server for content compression• Configuring the server for web publishing and content authoring using WebDAV	<i>Administrator's Guide</i>
Using programming technologies and APIs to do the following: <ul style="list-style-type: none">• Extend and modify Sun ONE Web Server• Dynamically generate content in response to client requests• Modify the content of the server	<i>Programmer's Guide</i>

Table 1 Sun ONE Web Server Documentation Roadmap

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 ONE Web Server	<i>Programmer's Guide to Web Applications</i>
Editing configuration files	<i>Administrator's Configuration File Reference Guide</i>
Tuning Sun ONE 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 ONE 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 Java™ 2 Enterprise Edition (J2EE™) 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 ONE Active Server Pages”](#)

Demonstrates how to enable a virtual server to handle dynamic content using Sun™ ONE 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/supporttraining/
```

- 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

Basic Tasks

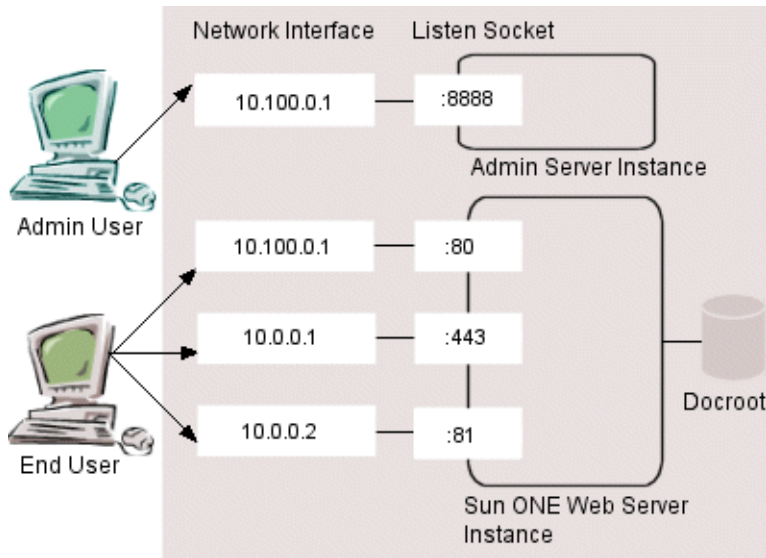
The Sun ONE Web Server is a secure and highly available server that you can use to host web sites. Each web site you create on Sun ONE Web Server is identified by a unique URL or address. A URL is a combination of a specific IP address, a port and a host name that follows the pattern: `http://www.sun.com/`.

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

When you install the Sun ONE Web Server 6.1 product, two server instances are installed by default:

- an Administration Server Instance
- a Web Server Instance.

This is shown in the figure below:



The Administration Server is created in the `server_root/https-admserv` directory and you use it to administer all of your Sun ONE 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.

In this section, we will learn how to perform the following tasks:

- [Start the Administration Server](#)
- [Access the Administration Server](#)
- [Start the Web Server Instance](#)
- [Access the Web Server](#)

Start the Administration Server

After you've installed Sun ONE Web Server, to access the Administration Server, you must first start it.

Let's assume that you installed the Administration Server at the default directory location, `/opt/SUNWwbsvr/`:

- To start the Administration Server on a UNIX machine, go to the `/https-admserv` directory:

- `cd /opt/SUNWwbsvr/https-admserv`
- and type `./start`
- Alternatively, you can type `./startconsole` in the server root directory:
 - `cd /opt/SUNWwbsvr`
 - `./startconsole`
- 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.

Access the Administration Server

After you've started the Administration Server, you can access it at any time by specifying the Administration Server address in a browser window:

`http://server_name:administration_port.`

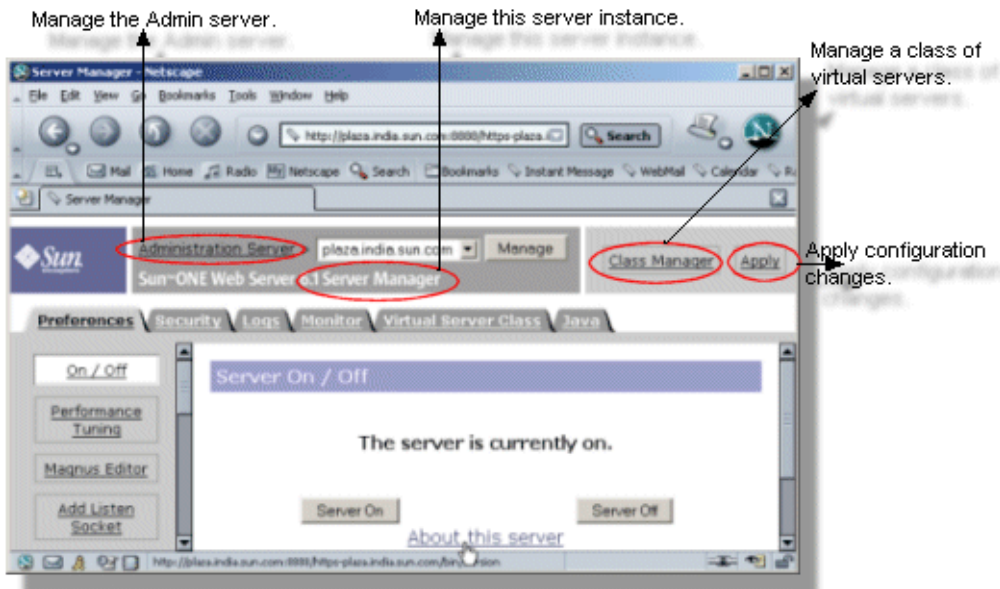
Let's assume that your server is called `acme` and during installation, you chose the default port `8888` as the port for the Administration Server to run on.

To launch the web-based Administration Server interface, open a browser window 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're ready to get started.

You can manage your web server(s), 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



For a complete description of the features each of these interfaces provides, refer to the Sun ONE Web Server 6.1 *Administrator's Guide*.

Next, let's turn to the default web server instance. This is turned off, by default, 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.

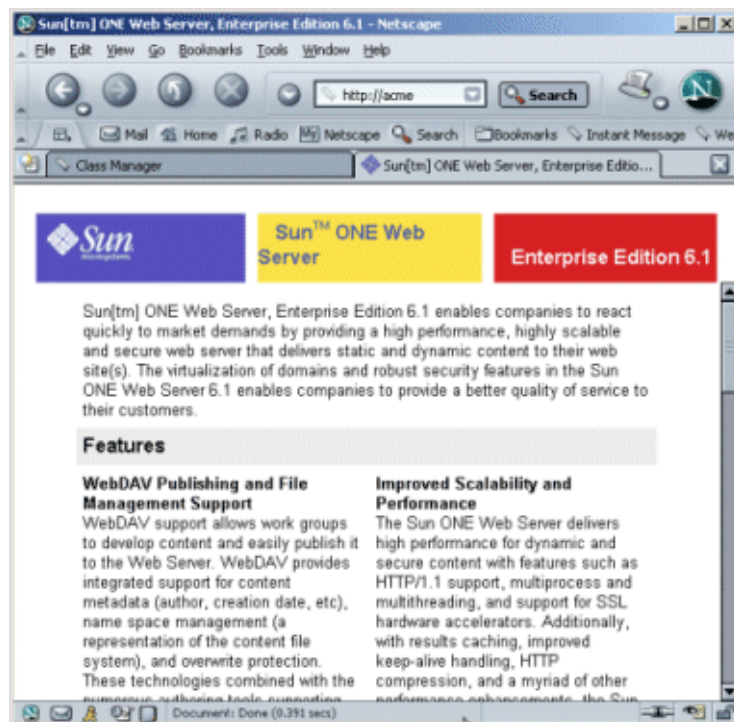
- Access the Administration Server interface in a browser and then click the Manage Server link. This brings up the Manage Server page where you can see your web server listed at the top of a drop-down menu. Click the Manage button. This takes you to the On/Off page in the Server Manager interface where 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

Let's walk through how you can access the web server interface next. Let's assume that during installation, you chose the default port, 80, as the port for your server, `acme`. To access its default home page, type the address in a browser window, as shown below.

Open a browser window and type `http://acme`.

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



The simplest way you can publish additional content on the site is by copying what you want to publish 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; doing so does not impair functionality in any way. For example, if you want to publish your own set of HTML pages, copy these into the `server_root/docs/` directory. By default, Sun ONE Web Server displays `index.html` as the server home page, which you can launch by typing `http://acme` in a browser window.

You don't however 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.

This is discussed in the section "[Create an Additional Document Directory](#)" in the next chapter "[Creating and Using Virtual Servers.](#)"

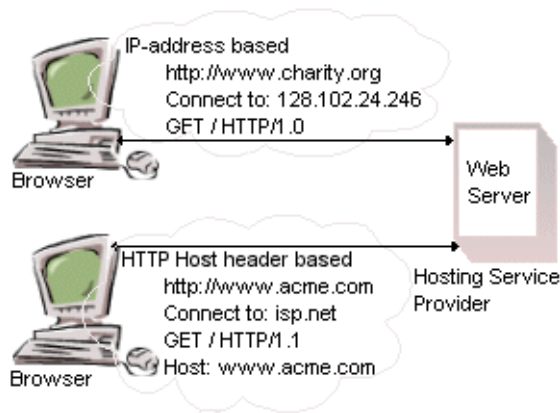
Creating and Using Virtual Servers

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 example, 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.

By using virtual servers, you can:

- **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.



As the figure above 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 ONE Web Server 6.1 supports both hardware (IP address-based) and software (HTTP Host Header-based) virtual servers.

When you install Sun ONE 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

With this in mind, let's set out to perform the following tasks:

- [Create a Listen Socket](#)
- [Create a Virtual Server Class](#)
- [Create a Virtual Server](#)
- [Create an Additional Document Directory](#)
- [Change the Document Root of a Virtual Server](#)

Let's say you want to create a virtual server, `hr.acme.com`. First, you'd need 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. We'll create a new listen socket in this example.

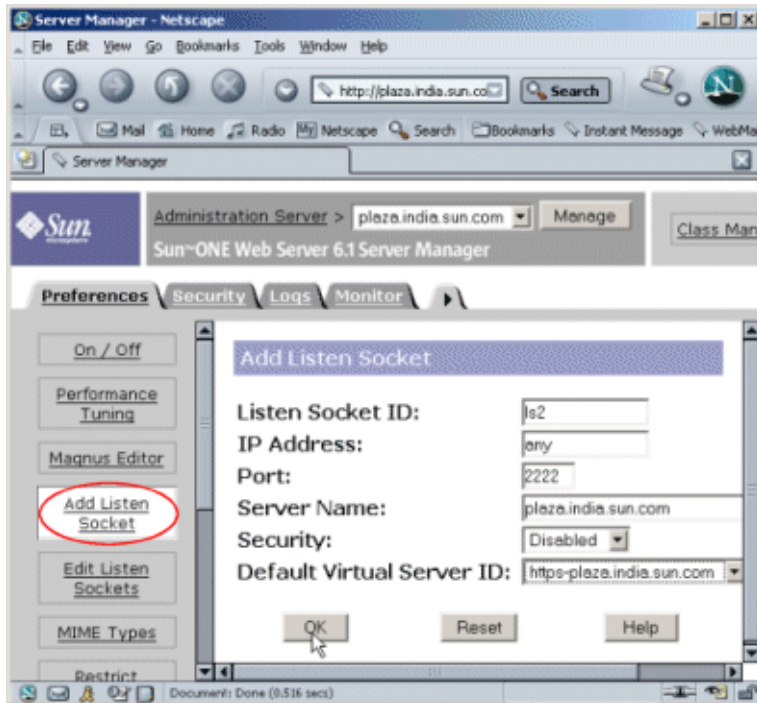
Create a Listen Socket

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 below:



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 don't have to configure each one separately. You could use the default virtual server class, `vsclass1`, that's created automatically when you install the server or you could add a new class.

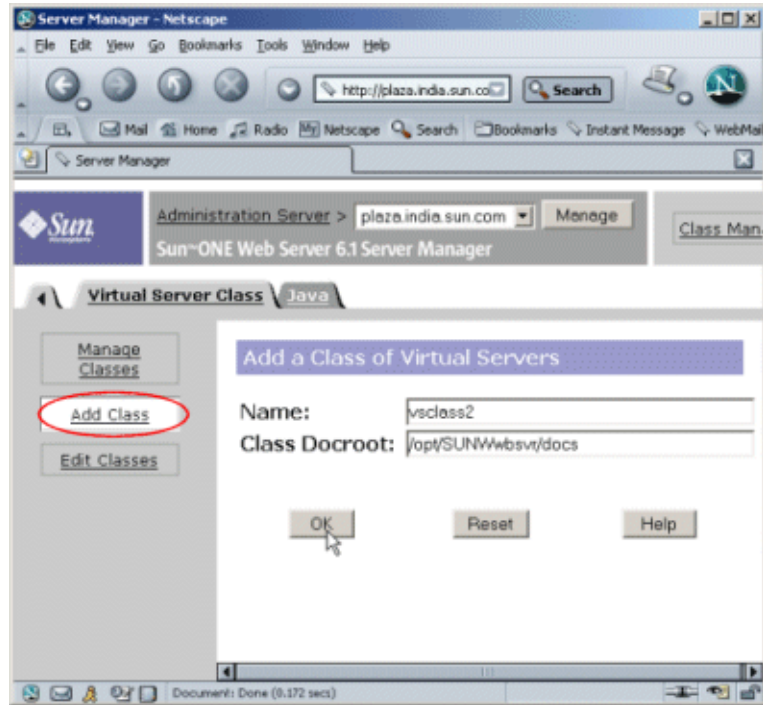
In this example, we will create a new virtual server class, `vsclass2`.

Create a Virtual Server Class

To add 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.



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

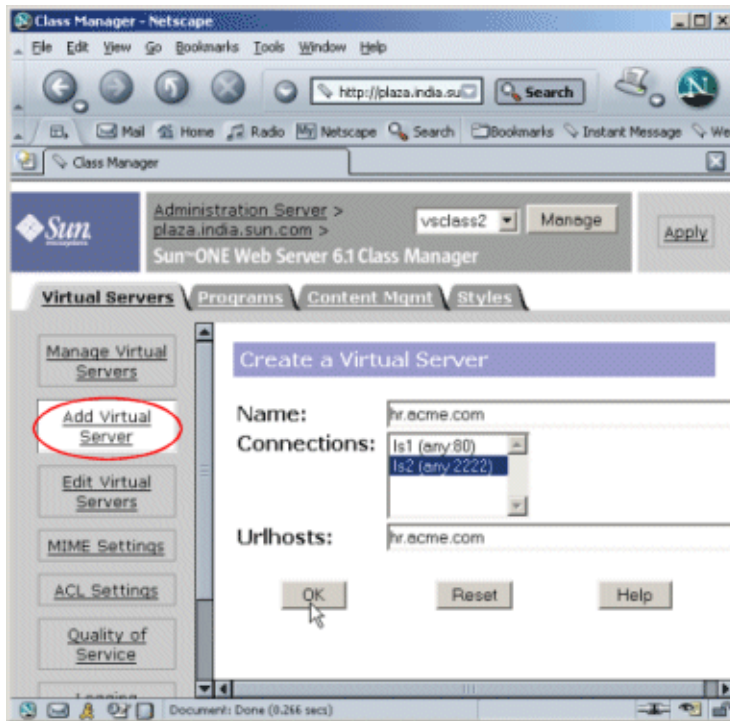
Next, we will create a new virtual server that will use the new listen socket we created, `ls2`, and will be managed by the virtual server class, `vsclass2`.

Create a Virtual Server

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.



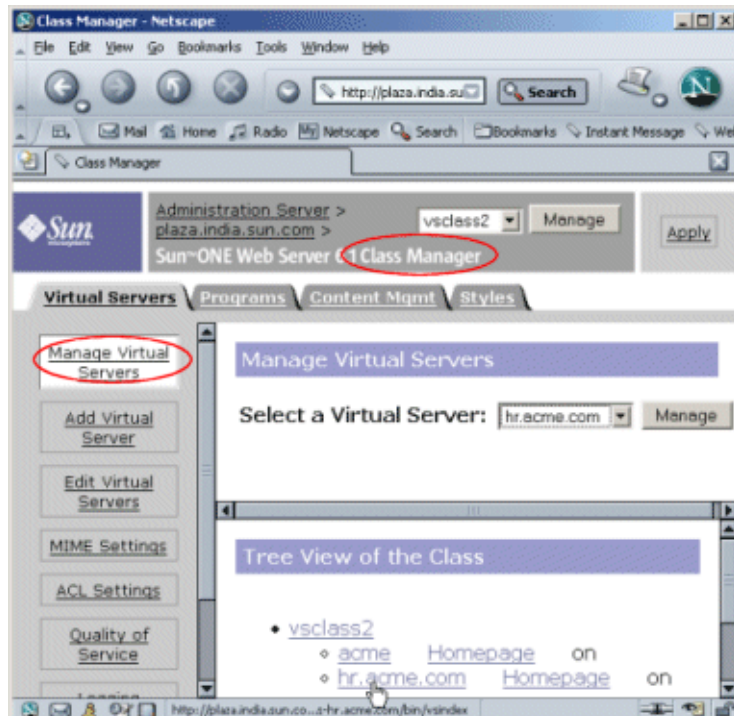
4. Click OK and then Apply to save and apply your changes.

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 of the settings for a specific server.



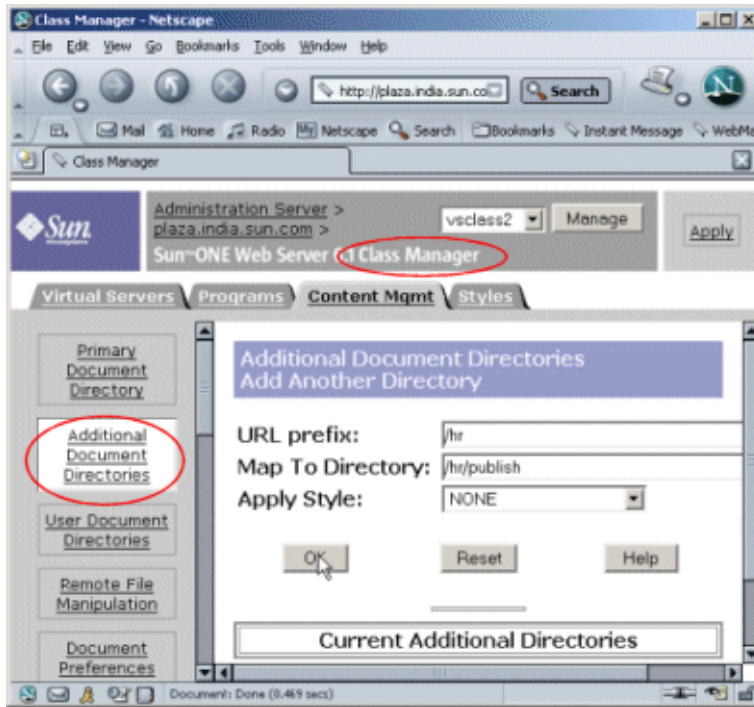
In the section “[Basic Tasks](#)”, we saw that the simplest way to publish content on the site was to put it in the docroot of the server. However, you don’t have to copy everything that you want to publish on your site into the docroot directory. You can also put content into any directory that you have mapped as an additional document directory. Let’s see how this is done.

Create an Additional Document Directory

Let’s say 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. We choose `/hr` as the URL prefix to map.



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 you've mapped.

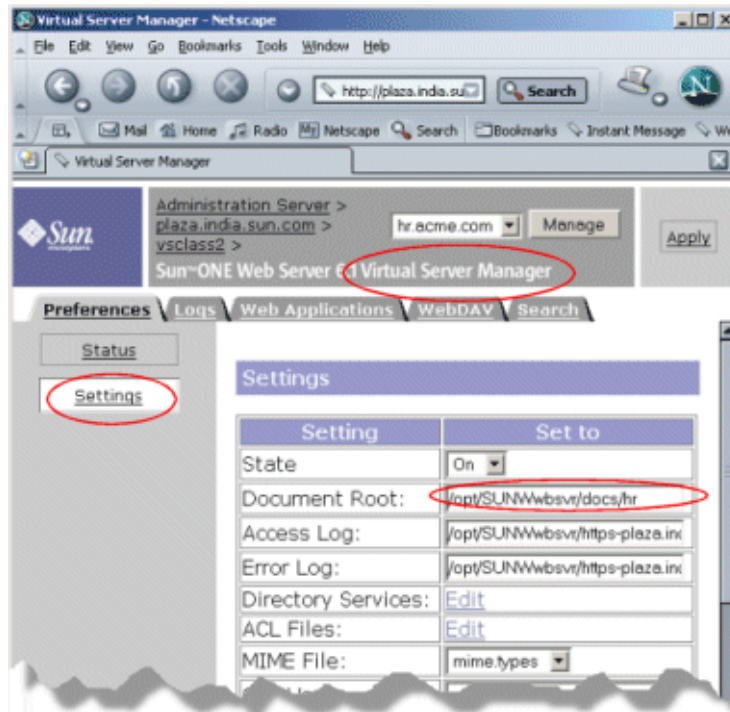
All the virtual servers in a virtual server class, by default, have the same document root. This is probably not what you need.

To explain, let's say you have two virtual servers `hr.acme.com` and `mis.acme.com` belonging to the same virtual server class. When a user types `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 users 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.

Change the Document Root of a Virtual Server

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.



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 differentiated services to different sets of end users, based on specific needs. The rest of this guide demonstrates 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 ONE Active Server Pages”](#)

Enabling Access Control

Every virtual server hosts one or more web sites. By default, everyone can access all the content on each web site. Sometimes, however, such unrestricted access might not be desirable and you might want to protect parts of your web site against 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. All users 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 users must supply a valid username and password before they can access your site. If you want to use user-based access control, you'd have to set up user and group accounts first.

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 on deploying Java web applications, you could also choose to 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 ONE Web Server 6.1 *Programmer's Guide to Web Applications*.

Let's walk through a simple exercise to see how a native ACL is configured. In this example, we will use combination of host-based and user-based access control. We will perform the following tasks:

- [Create a Directory Service](#)

- [Add a User](#)
- [Specify a Directory Service for a Virtual Server](#)
- [Set Access Control](#)

Let's assume that we want to allow access to all files under `/hr/publish/manager` to a user named "manager" at Acme Corp.

Since we shall be using user-based access control (in addition to host-based access control), we first need to create a **directory service**.

Create a Directory Service

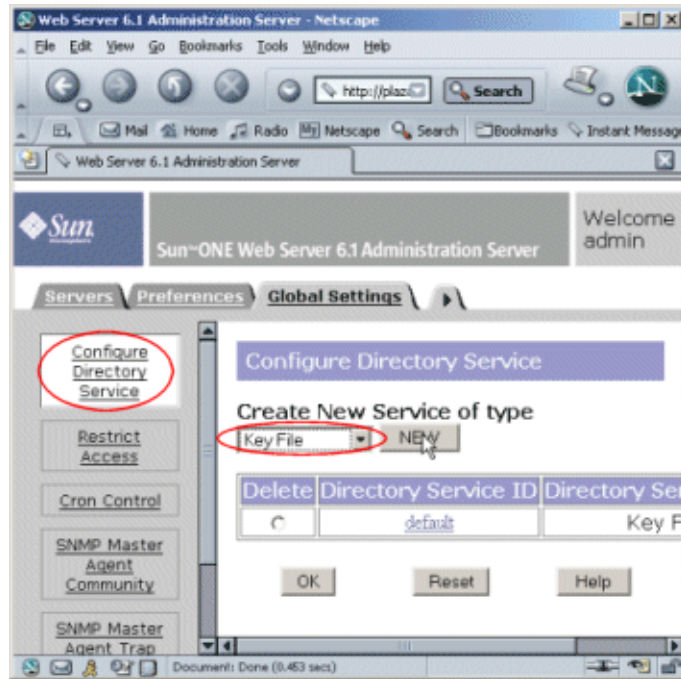
A directory service allows you to authenticate and authorize users and groups. We can configure a directory service in one of the following ways:

- 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.

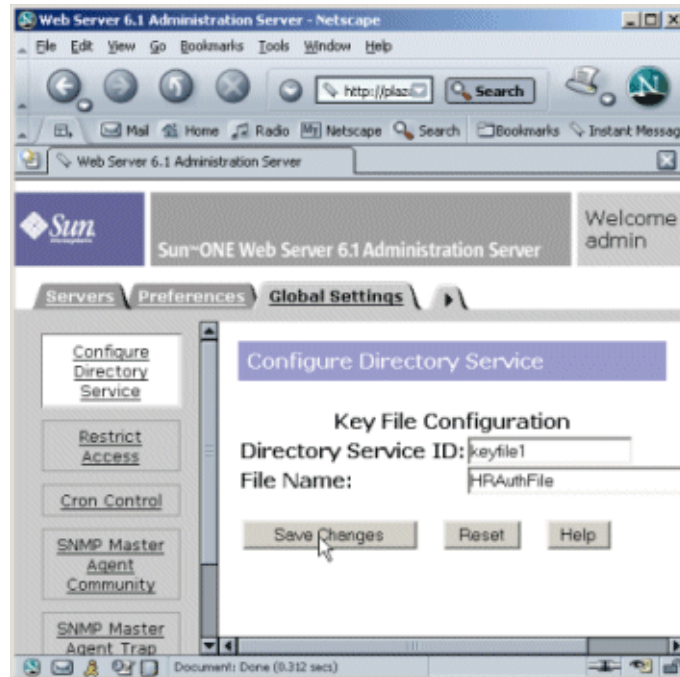
1. Access the Administration Server and choose the Global Settings tab.
2. Click the Configure Directory Service link.

- 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.



- Click New.

5. Specify “keyfile1” as the Directory Service ID and “HRAuthFile” as the name of the file, as shown below:



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

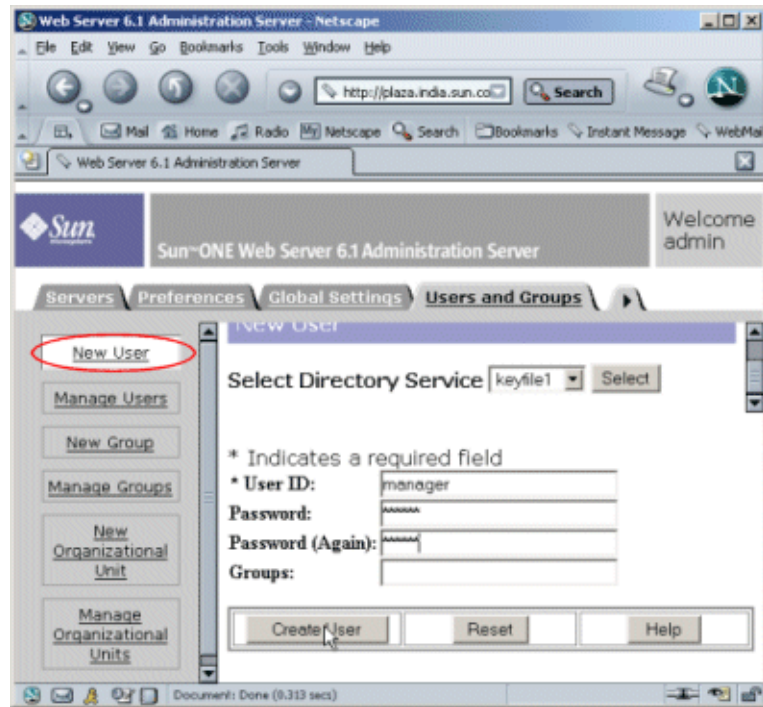
Now that we’ve created a file-type database, “HRAuthFile”, we need to add users to it.

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:



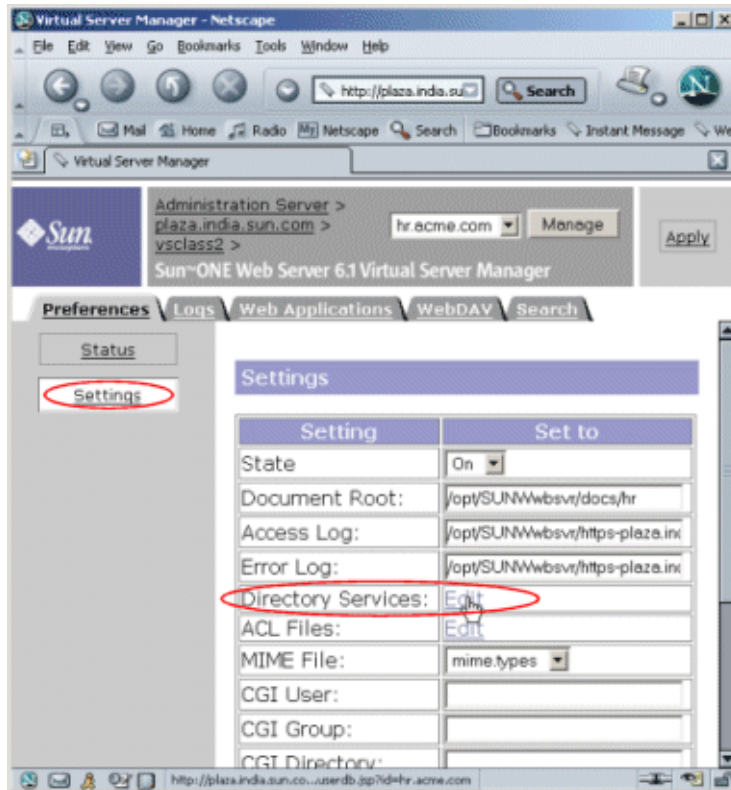
5. Click Create User.

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

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

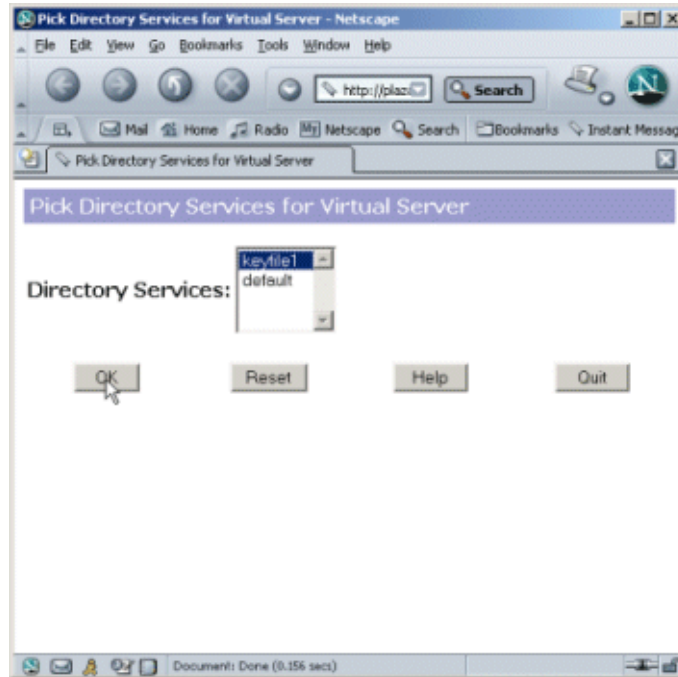
Specify a Directory Service for a Virtual Server

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



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:



4. Click OK and then Apply to save and apply your changes.
We're now ready to specify the required access control rules.

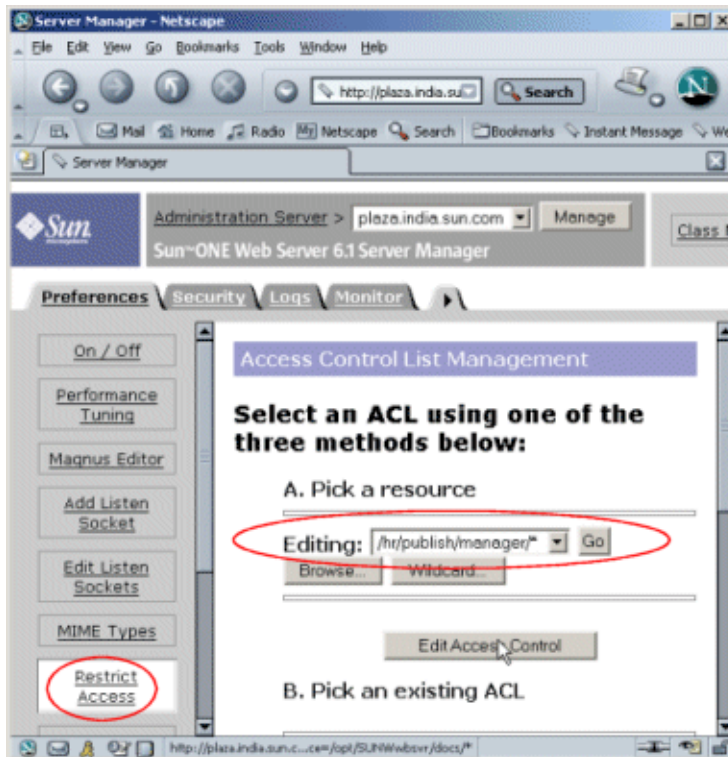
Set Access Control

We start out 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.

4. Click OK.

The Access Control List Management Page appears.

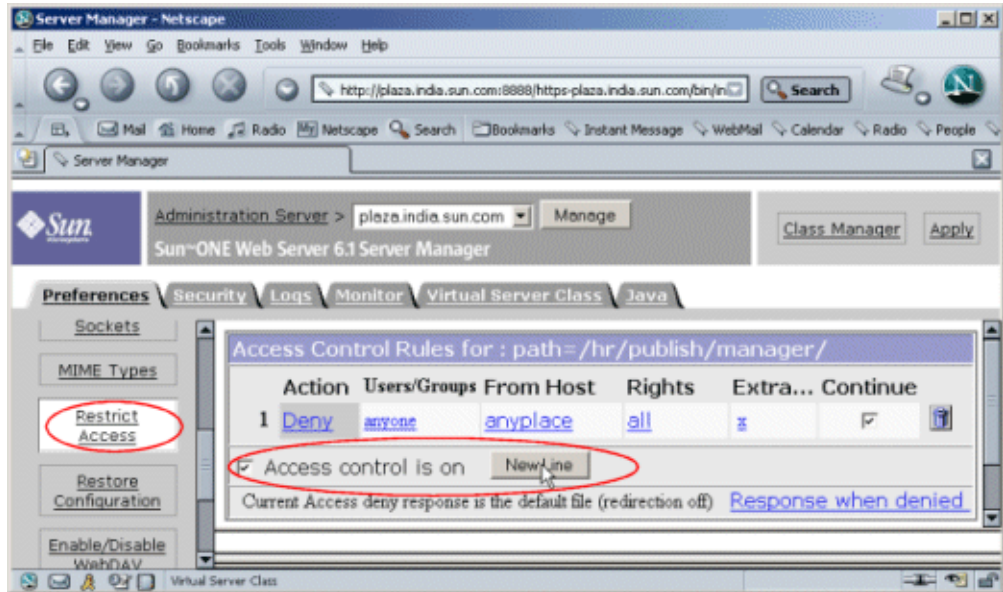


5. Select Pick a Resource and then specify the following wildcard pattern to control access to all files in the /hr/publish/manager directory :

/hr/publish/manager/*

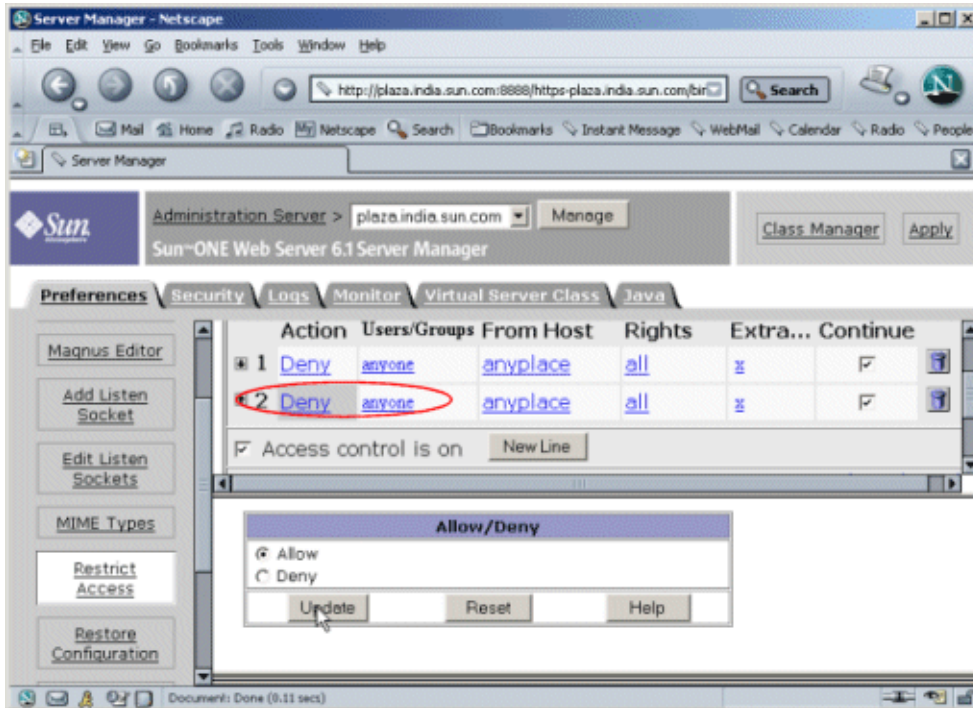
6. Click the Edit Access Control button.

7. Check the “Access control is on” checkbox, if not already selected, and click the New Line button.



8. Click on Deny in the second row of the Action column.

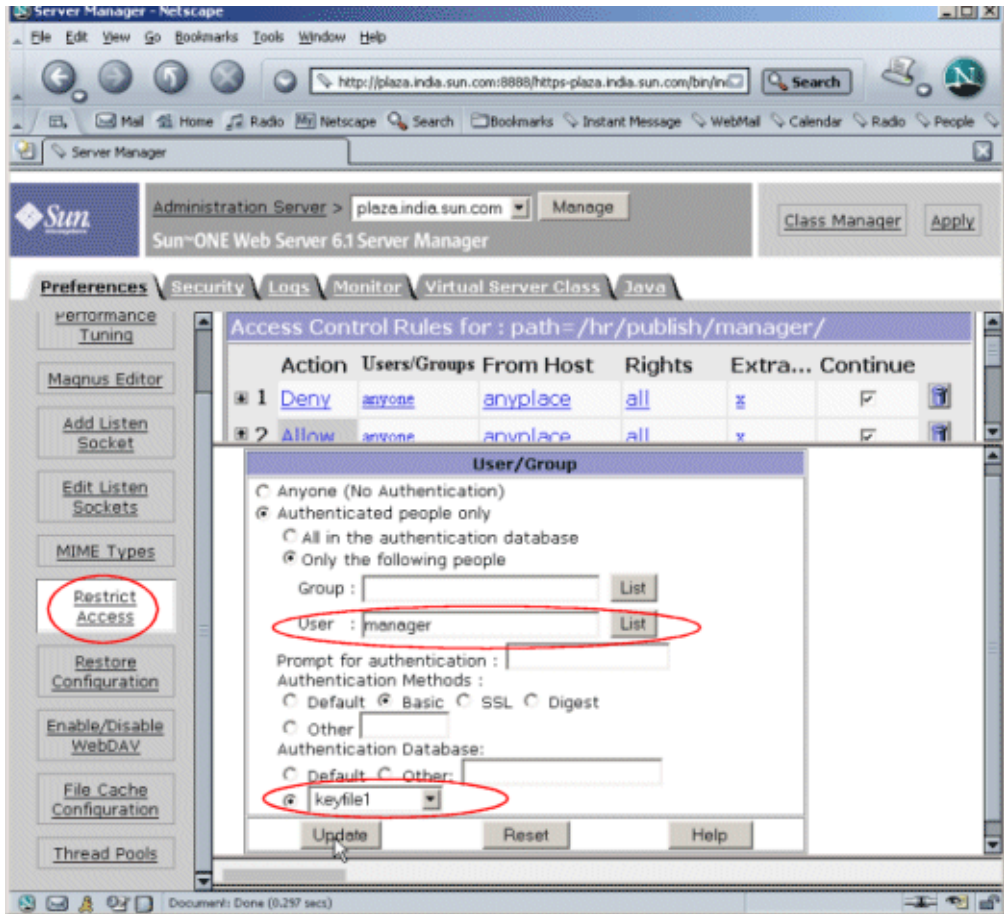
This opens the Allow /Deny page in the lower frame:



9. Click Allow, if it isn't already selected by default, and then click Update.

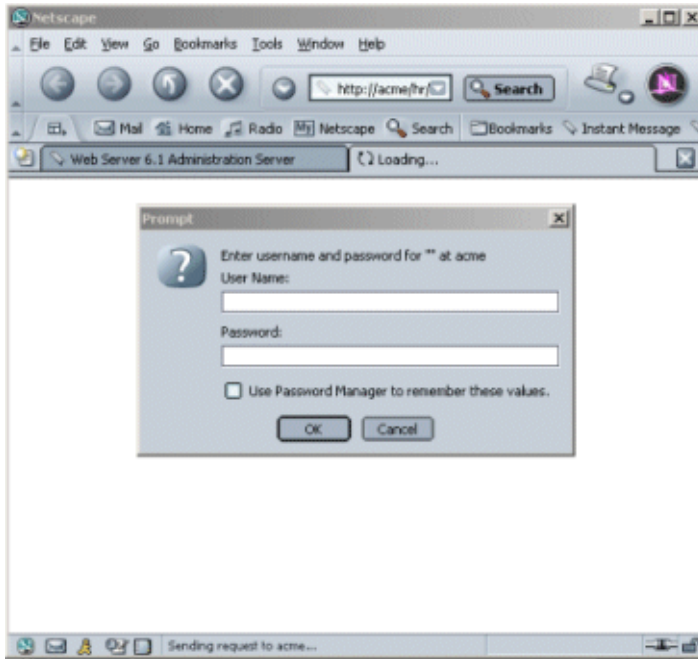
10. Click on "anyone" in the Users/Groups column, in the top frame.

11. 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, as shown below



12. Click Update.
13. Click Submit to store the new access control rules in the ACL file.
14. Click Apply Changes and save and apply the changes you've made.

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



Publishing Content

Sun ONE 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've created a WebDAV collection on your virtual server (we'll see how that's done a little further on), users no matter where they are located, can access the documents from the collection as easily as if they were accessing documents on a local file system.

A complete WebDAV transaction involves a server that supports WebDAV, such as Sun ONE 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.

In this section, we will set up both aspects of the WebDAV transaction:

- [Set Up the Server to Handle WebDAV](#)
- [Set Up a Client to Use WebDAV](#)

Set Up the Server to Handle WebDAV

Clients can author documents and publish them on your server, 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.

Let's walk through some of the basic tasks involved.

- [Enable WebDAV](#)
- [Create a WebDAV Collection](#)

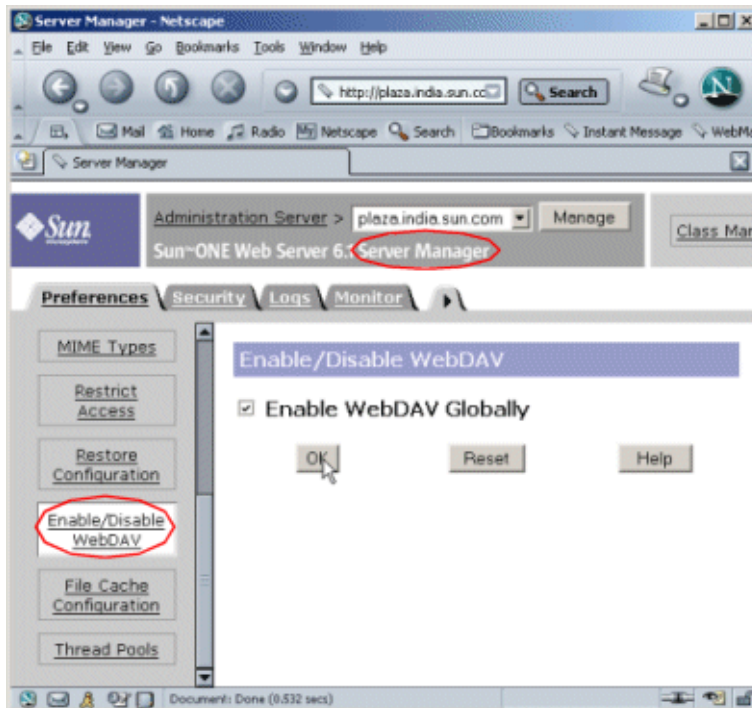
Let's assume we have a set of documents located in a folder called `/info` in the docroot of your virtual server, `acme`. You want the HR group in Acme Corp. to be able to work remotely with this folder.

Enable WebDAV

First you need to enable WebDAV at two levels: globally, at the server instance level and at the virtual server class level. (By default, WebDAV is disabled.)

To enable WebDAV globally:

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

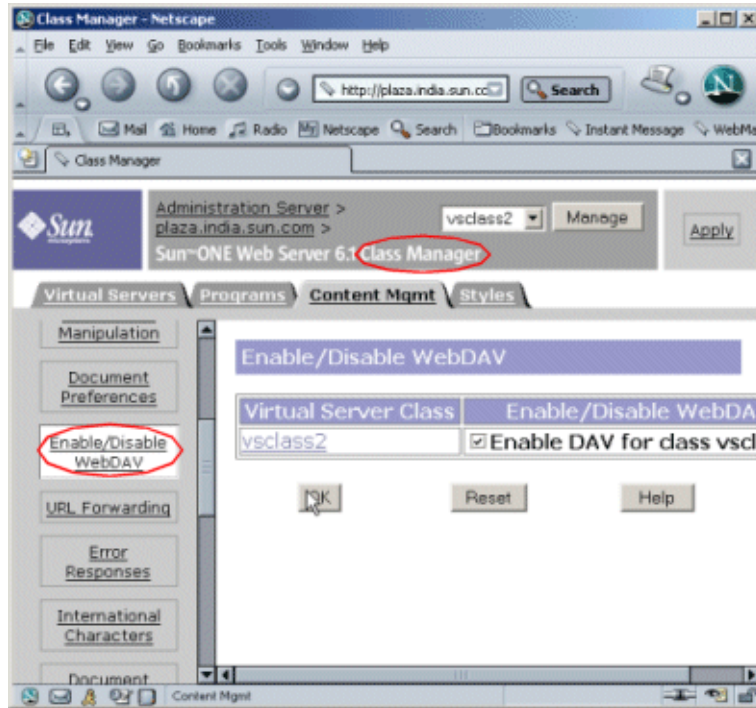


2. Click the Enable WebDAV Globally checkbox.
3. Click OK and then Apply for the changes to take effect.

To 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.



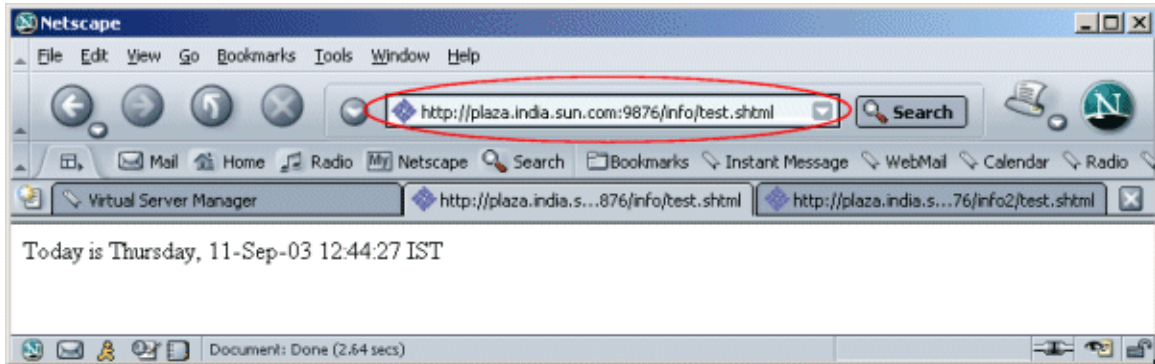
3. Check the Enable DAV checkbox corresponding to vsclass2.
4. Click OK.

You're now ready to create the WebDAV collection.

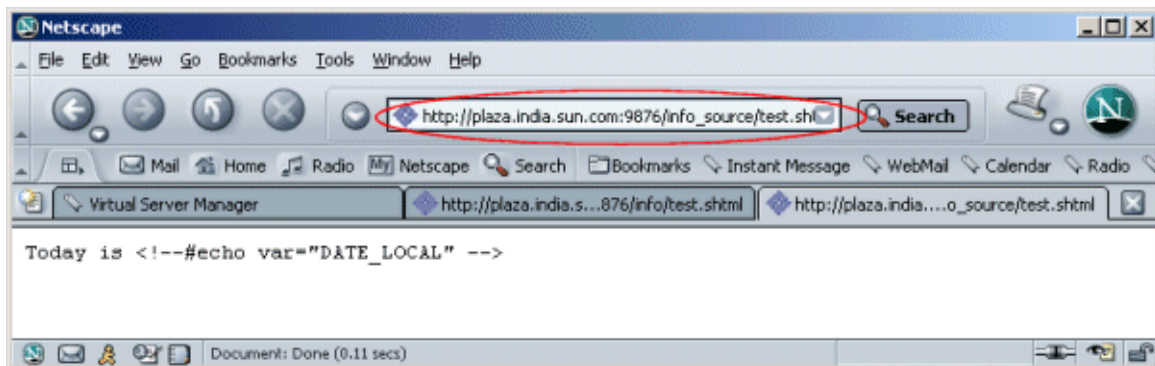
At a minimum, creating a collection involves nothing more than specifying the path to a directory that you want to enable WebDAV on. Now, if this directory contains only static data (such as HTML pages or images), you need to specify only the **URI** path to the directory. However, if the directory also contains dynamic content (such as JSP, SHTML, or CGI programs), you must also specify a **source URI**. What's the difference between specifying a URI and specifying a source URI for a collection? Well, to understand the difference, let's consider the following example:

Let's assume we have a dynamic page, `test.shtml` in the collection. This page contains a script that displays today's date on the browser.

Let's say that 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, the figure below shows how it appears on a browser window:



Next, let's say you 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:



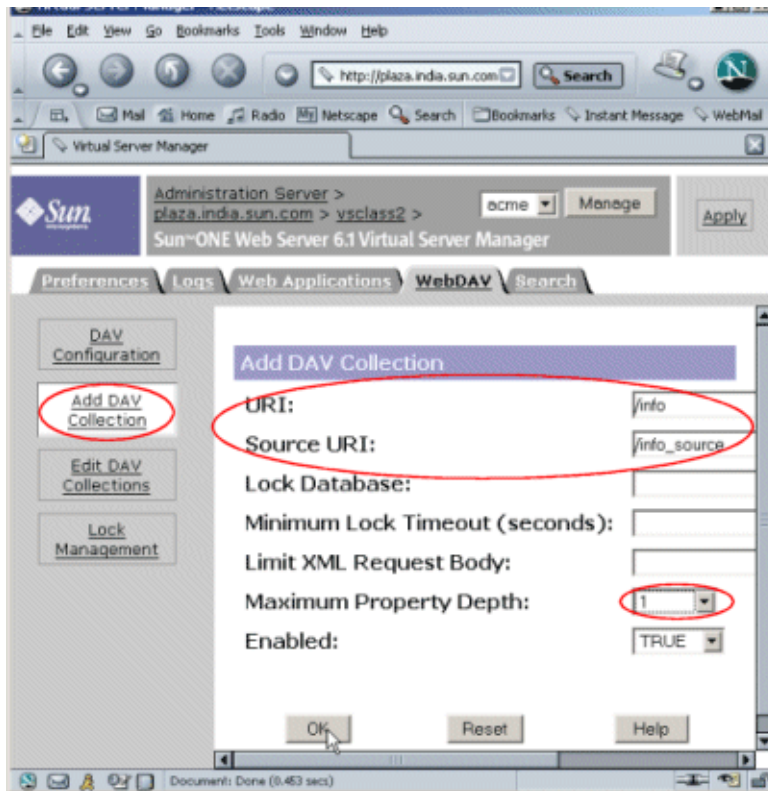
By specifying a source URI for a collection, you thus expose the source of a resource to clients. Exposing the source obviously has security implications and you are recommended to set access control on directories that you map source URIs against. This is discussed in the section [Chapter 3, "Enabling Access Control"](#).

Let's create a collection next, and specify both a URI and also a source URI so that the collection can expose the source of dynamic content.

Create a WebDAV Collection

To create your 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.



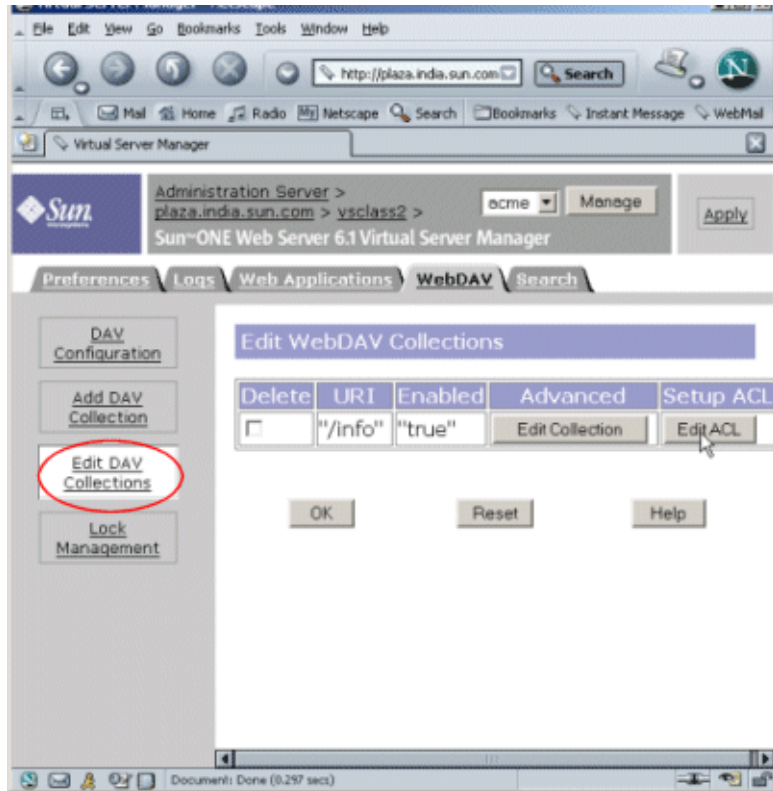
4. Set the Maximum Property Depth value to either 1 or infinity.
5. Click OK and then Apply to save and apply your changes.

You might want to specify other parameters for the collection in order to fine-tune the level of WebDAV support. You might for example, want to 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 could do so either at the time of creating the collection or at any time after you've created the collection using the Edit DAV Collections page.

Once you're 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. A couple of points of note here:

- 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. In the section [Chapter 3, "Enabling Access Control"](#), we discuss how you can enable access control and security.

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



Now that we've set the server up to handle WebDAV requests, let's turn to the client-side tasks next.

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 ONE 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, you'd specify the URL for the client to make a connection with the server:

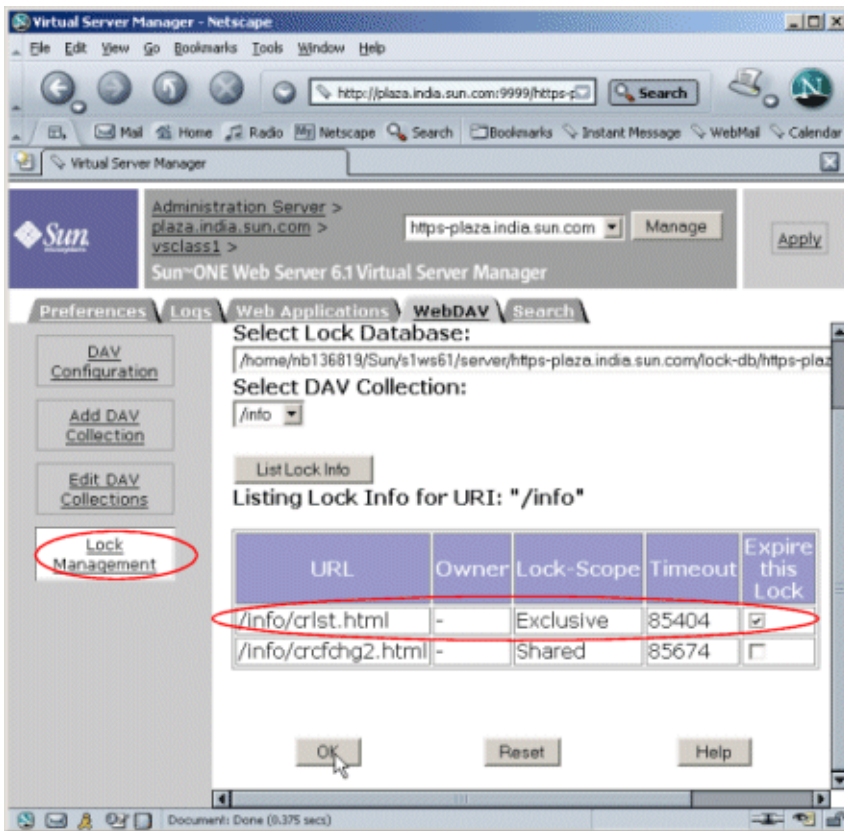
```
http://acme:2222/info
```

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

You could then perform several operations on the listed resources. For example, you could create 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, you would use the WebDAV client software to lock it to prevent others from simultaneously working on the same file and overwriting your changes.

Sun ONE 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 below:



The WebDAV functionality is discussed at length in the Sun ONE Web Server 6.1 *Administrator's Guide*. Refer to it for more information on configuring and fine-tuning your WebDAV resources.

Making Content Searchable

In this chapter we'll see how we can make content searchable on a virtual server. Sun ONE Web Server 6.1 includes a search engine that allows users 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.

Before we can enable search on a virtual server, we must first 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 ONE 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.

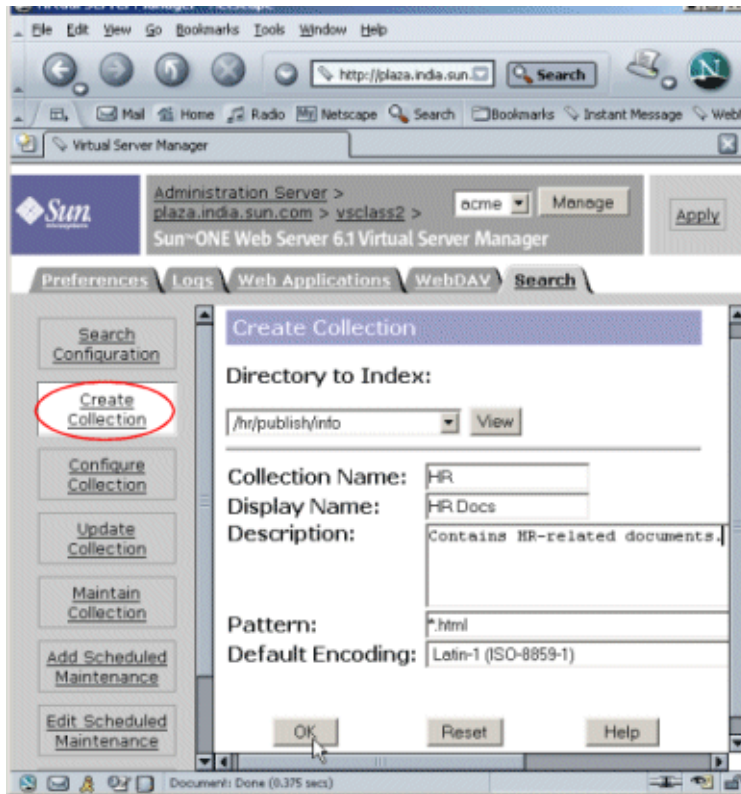
Next, because searches require a database of searchable documents (called a collection) against which users can search, you need to create this database. At a minimum, you must have at least one collection configured on your server. You can however create as many collections as required.

Let's walk through a few basic search-related tasks:

- [Create a Search Collection](#)
- [Update Your Collection](#)
- [Enable the Search Application](#)
- [Customize the Search Application](#)

Create a Search Collection

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



In the figure above, we've specified certain parameters that will 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 [“Create an Additional Document Directory.”](#)
- “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 we'd left this field 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.

- We've left the Encoding field 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, say, "Shift_JIS."

You need to save and apply your changes after you're done for the changes to take effect.

You may 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.

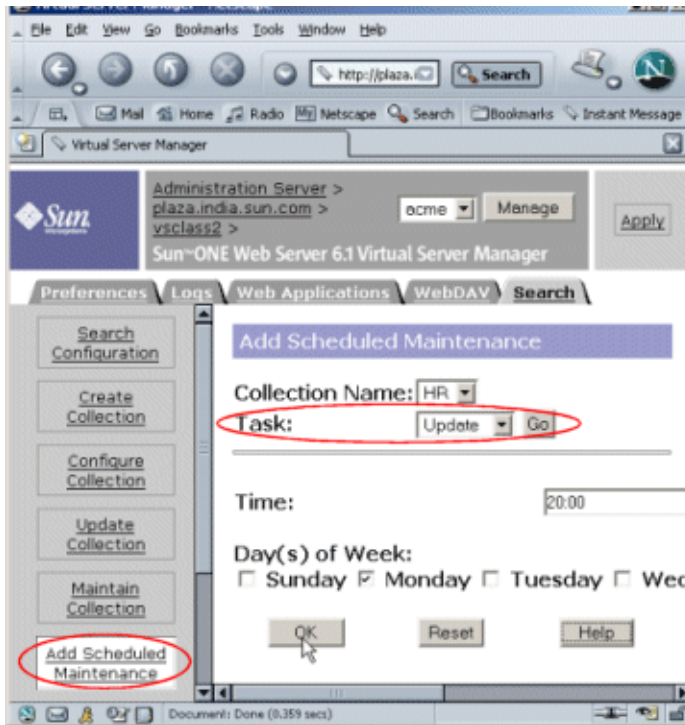
Now that we've created our collection, we need to keep some additional points in mind.

Update Your Collection

When you add new documents to the `/hr/publish/info` folder, you must update the "HR" collection to index the new documents into the collection. You can do so manually using the Update Collections page.

However, if your site is constantly being updated, you might want to 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.

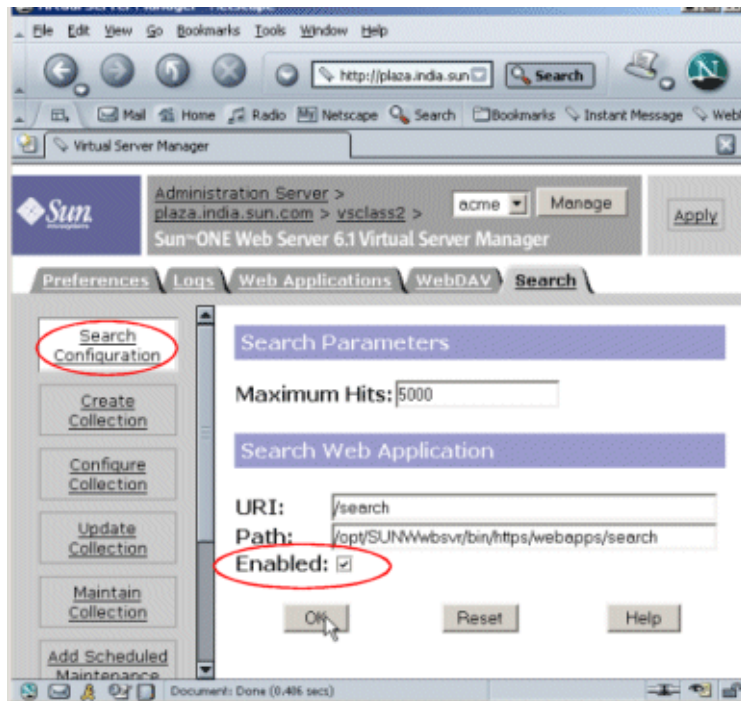
To automate maintenance tasks on collections, use the Add Scheduled Maintenance link in the Search tab of the virtual server. Choose Update as the type of task you want to automate, and specify other parameters as appropriate, for example, the time of day to perform the task, the day(s) of the week to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



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're planning on using the default search application, all you need to do is to click the Enabled checkbox on this page and use the default values for the rest of the parameters.



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

<http://acme/search>



Customize the Search Application

Finally, you may choose to customize the search application to suit your specific needs. You can use the JSP tag libraries included with Sun ONE 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 Customizing Search Pages in the Sun ONE Web Server 6.1 *Administrator's Guide* provides examples of customizing the search page, for example, 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, refer to the Sun ONE Web Server 6.1 *Programmer's Guide to Web Applications*.

Running Java Web Applications On Your Server

Sun ONE Web Server 6.1 allows you to create and deploy web applications that extend your server dynamically. Sun ONE 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.

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.

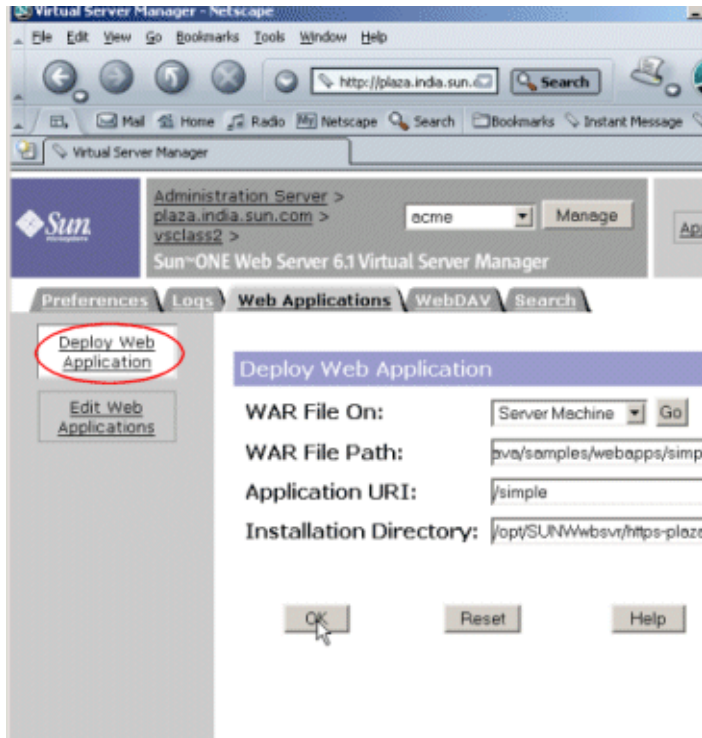
To deploy a web application using the web-based Sun ONE Web Server interface you need to specify where the web archive (WAR) file for the application is located.

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

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

To deploy this application:

1. Access the virtual server and click the Web Applications tab

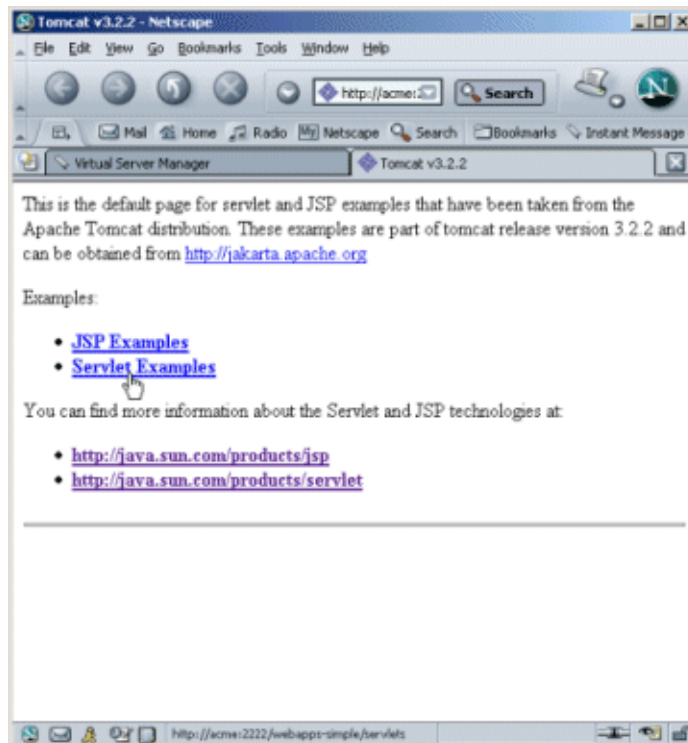


2. Because the WAR file resides on the server, select Server Machine from the “WAR File On” drop-down list and click the Go button.
3. Specify the following:
 - The absolute path to the sample WAR file, webapps-simple.war:
`<install_root>/plugins/java/samples/webapps/simple/webapps-simple.war`
 - The application URI:
`/simple.`

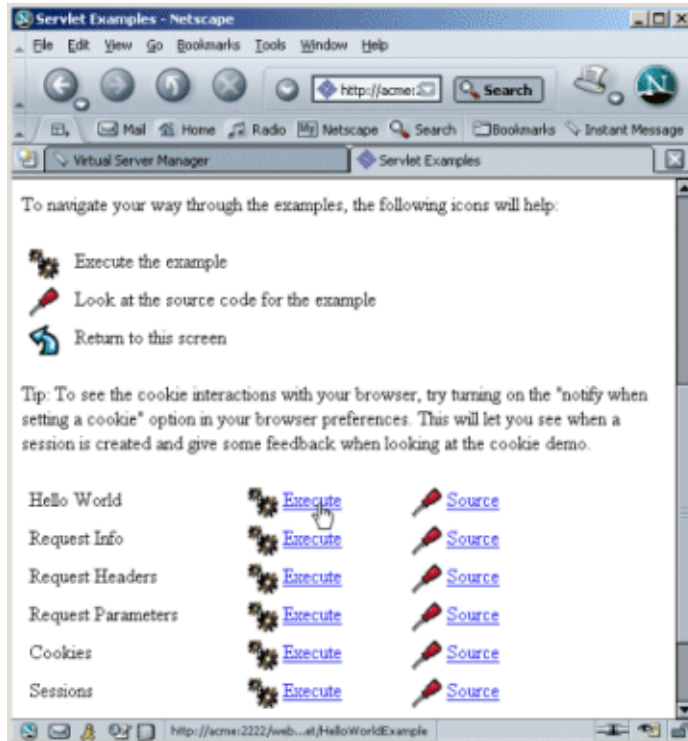
You needn't make any changes to the default installation directory

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`



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 below:



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 ONE Web Server. In the following section, “[Setting Up Database Connectivity](#)”, we will use of these sample applications, the `jdbc-simple` sample, to demonstrate how you can set up database connectivity and use Java resources in web applications.

Setting Up Database Connectivity

Web applications leverage the benefits of the component-based Java™ 2 Platform, Enterprise Edition (J2EE™) technology, which include scalability and speed.

Web components, Java Servlets or JSP pages, provide the dynamic extension capabilities for Sun ONE 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.

In this section, we will work with a sample application that uses the Java™ Database Connectivity (JDBC™) API to record greetings to a database.

In this section you will 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](#)
2. [Start the Database Server](#)
3. [Create a Database and Table](#)
4. [Define the JDBC Connection Pool](#)
5. [Define the JDBC Resource](#)

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, we use the PointBase JDBC driver.

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

1. [Download and Install PointBase Server and Client Products](#)
2. [Add PointBase Type 4 JDBC Driver to the Web Server's Classpath](#)

Download and Install PointBase Server and Client Products

Download the PointBase evaluation software from <http://www.pointbase.com/>.

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

Add PointBase Type 4 JDBC Driver to the Web Server's Classpath

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 the web server's configuration:

1. Access the web server instance's Java tab -> 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.

2. Once you've changed the Classpath Suffix field, apply the changes and restart the web server instance.

Start the Database Server

The database server can be easily 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, you will see the following text in either a command window or at the UNIX terminal prompt:

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

Create a Database and Table

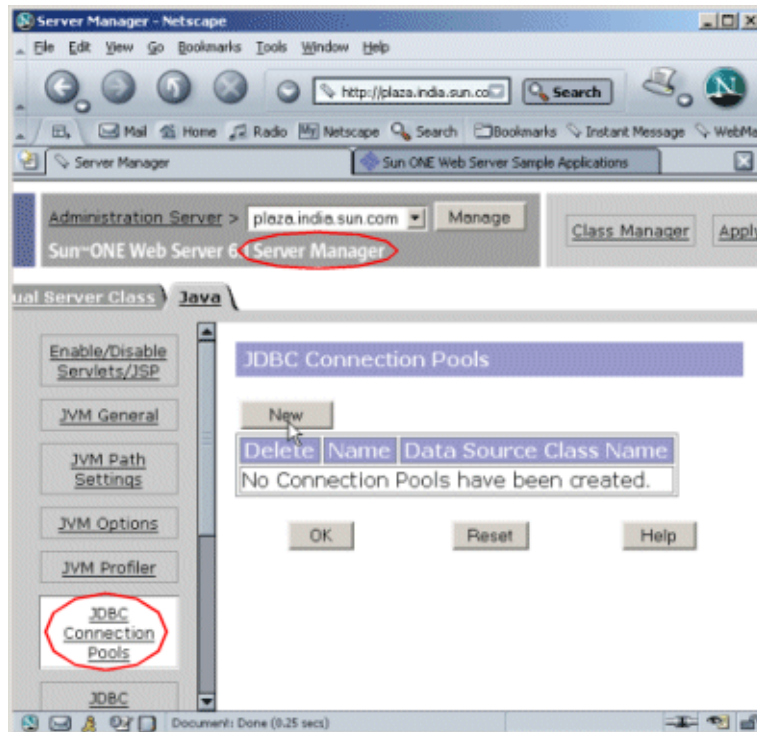
For information about creating and maintaining a database using Pointbase, refer to the database documentation. After you have created a database, you need to use the following script to create the table 'Greeting', which will record application entries:

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

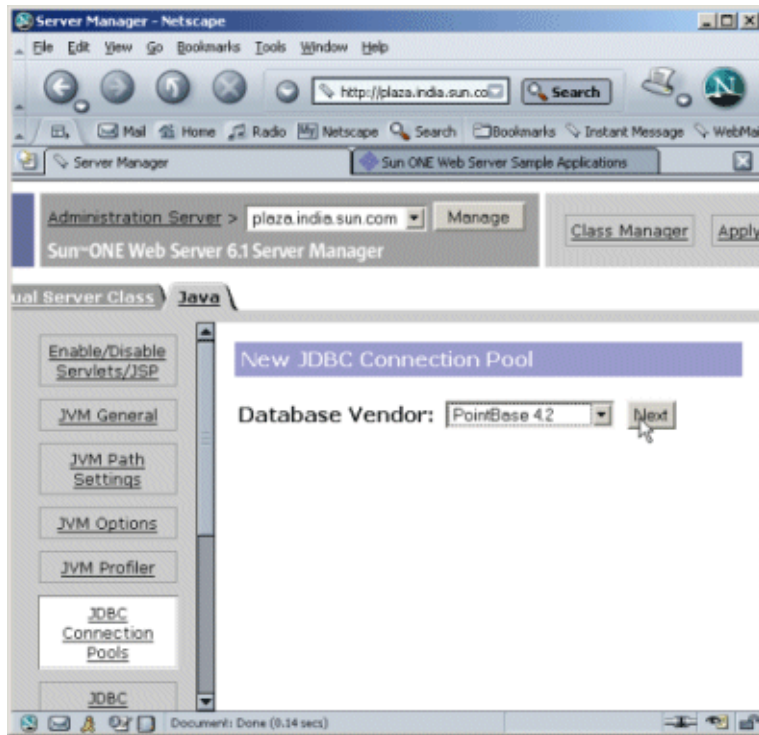
Define the JDBC Connection Pool

Before running the sample application, you need to define a suitable JDBC connection pool that maps to the PointBase database server and 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. (You might have to start the administration server if it is not running).
2. Select the Java tab in the server instance and click JDBC Connection Pools

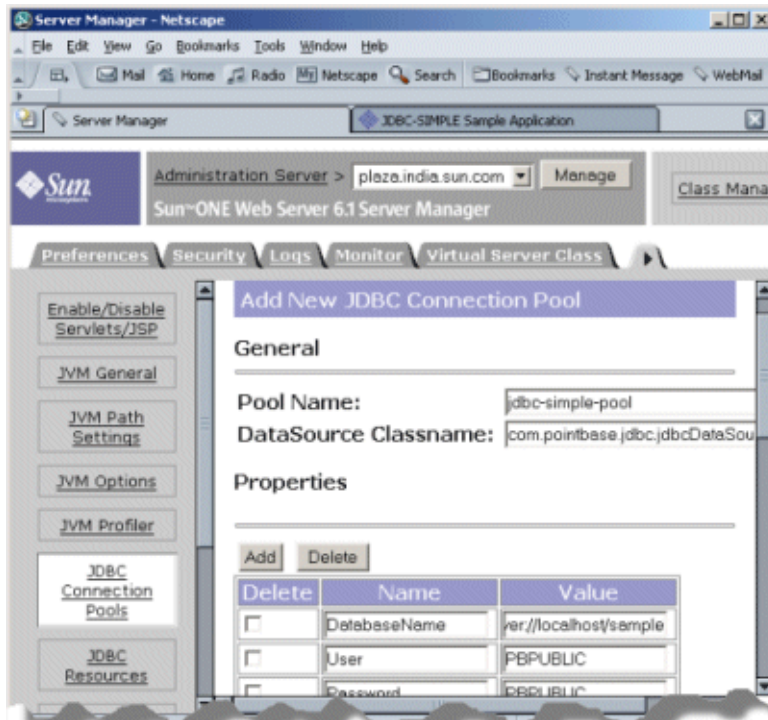


3. Click New, and select the JDBC driver, as shown below:



4. In the Add New JDBC Connection Pools page, we've specified the following settings and left the rest unmodified:
 - o jdbc-simple-pool as the name of the new connection pool.
 - o jdbc:pointbase:server://localhost/sample as database name
 - o PBPUBLIC as the user name
 - o PBPUBLIC as the password

- Transaction Isolation set to read-committed



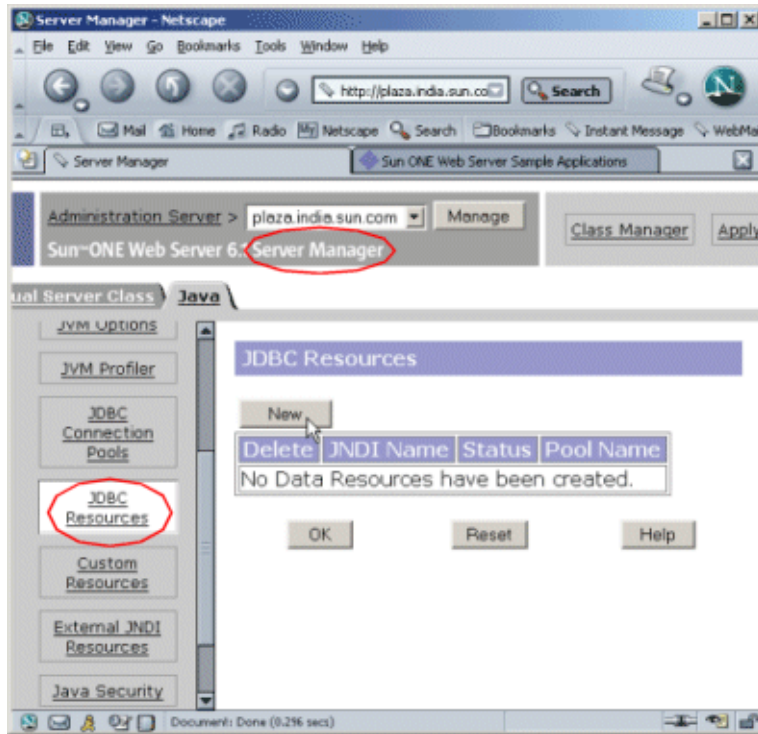
5. Click OK and then Apply to save and apply your changes.

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.

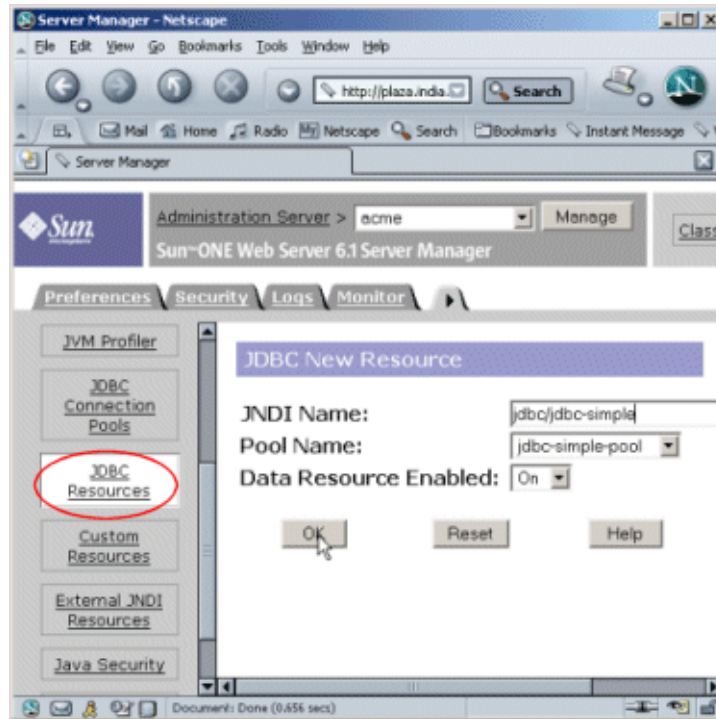
1. Under the Java tab of the server instance, click the JDBC Resources link.

2. Click the New button to define a new resource entry.



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 the OK button to define the resource:



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

After you've set up database connectivity, you're 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 the section [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. In this section, we will look at using CGI programs. In the following section [Chapter 9, “Using Sun ONE Active Server Pages”](#), we will use Sun™ ONE Active Server pages to create dynamic content.

For details about using SHTML tags, see the Sun ONE Web Server 6.1 *Programmer’s Guide*, and for details on using NSAPI, see the Sun ONE Web Server 6.1 *NSAPI Programmer’s Guide*.

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

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

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

Let's assume we 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.

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 web server's server root directory), and save your changes:

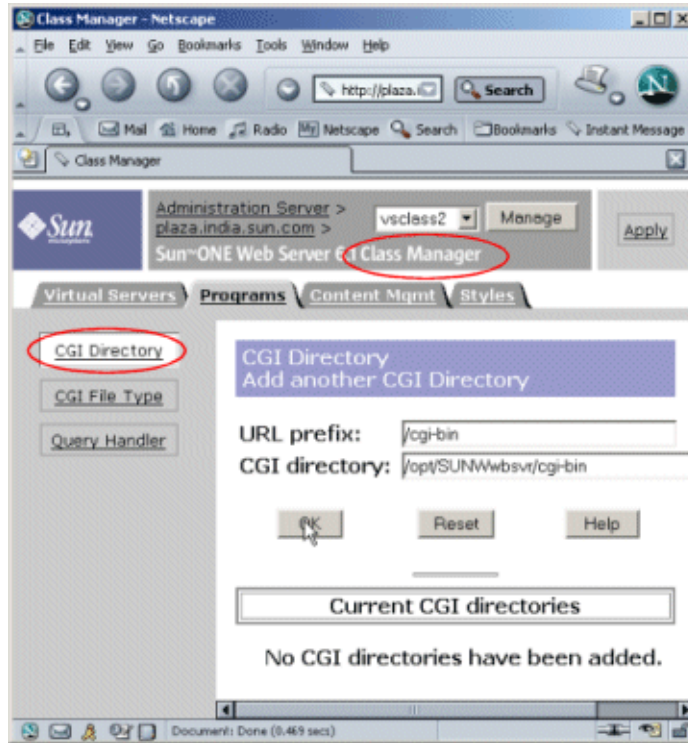
```
#!/<server_root>/bin/https/perl/perl
print "Content-type:text/html\n\n";
print "Hello World!";
```

2. Next, we need to create a directory for this program. By convention, this directory has come to be known as the `cgi-bin` directory though you can name it whatever you like.

```
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 the Programs tab.



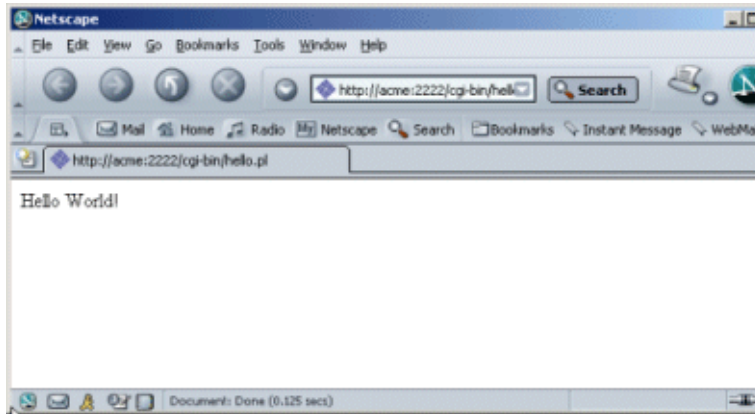
5. Click the CGI Directory link if the server is running on UNIX or on Windows, the WinCGI Directory link. Specify:
 - o a URL prefix (to which clients will send requests). In our example, this is set to /cgi-bin
 - o the CGI directory. In our example, /opt/SUNWwbsvr/cgi-bin.

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 below:



Using Sun ONE Active Server Pages

Sun ONE Web Server 6.1 supports Sun™ ONE 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 ONE Active Server Pages Server.

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

```
http://www.sun.com/software/chilisoft/index.html
```

Note that you need to log in as root to install Sun ONE Active Server Pages.

When you install Sun ONE 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 ONE Active Server Pages, type the specified URL on a browser window.



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

In this section, we will deploy a simple application, `index.asp`, which displays the current date, and run it off the Web Server.

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. Next, we need to 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 the “Add a new application” link.



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.



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.



Index

A

- About this guide [5](#)
- Access the Administration Server [13](#)
- Access the Web Server [15](#)
- Add a User [30](#)
- Add New JDBC Connection Pools page [62](#)
- Additional document directories [16](#)
- Administration interface
 - more information about [7](#)
- Administration Server [13](#)
- Administration Server Instance [11](#)
- Allow /Deny page [36](#)
- ASP [67](#)
 - Downloading Sun ONE ASP [71](#)
 - Using Sun ONE Active Server Pages [71](#)

B

- Benefits of virtual servers [17](#)

C

- CGI
 - cgi-bin directory [68](#)
 - Perl interpreter [67](#)
 - Using CGI [67](#)
- cgi-bin directory [68](#)

- Change the Document Root of a Virtual Server [25](#)
- Class Manager [13, 22](#)
- Classpath [60](#)
- Create a Directory Service [28](#)
- Create a Listen Socket [19](#)
- Create a Search Collection [49](#)
- Create a Virtual Server [21](#)
- Create a Virtual Server Class [20](#)
- Create a WebDAV Collection [43](#)
- Create an Additional Document Directory [23](#)

D

- Default search web application [49](#)
- Define the JDBC Resource [63](#)
- Deploying Java Web Applications On Your Server [55](#)
- Difference between URI and Source URI [41](#)
- directory service [28](#)
- document root [11](#)
- Documentation roadmap [5](#)

E

- Enabling Access Control [27](#)
- 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 [59](#)

J

- J2EE [59](#)
- Java Database Connectivity (JDBC) [59](#)
- Java security constraints [27](#)
- JDBC [59](#)
- JDBC Connection Pool [61](#)
- JNDI Name [65](#)
- JSP tag libraries [54](#)

K

- Key File [29](#)

L

- listen socket [18](#)

M

- Making Content Searchable [49](#)

N

- native access control [27](#)
- NSAPI [67](#)

P

- Perl interpreter [67](#)
- PointBase Server [60](#)
- Publish additional content [15](#)

S

- sample web application [65](#)
- Search
 - Add Scheduled Maintenance [51](#)
 - collection [49](#)
 - collection name [50](#)
 - create a collection [49](#)
 - customizable search tags [54](#)
 - customize the Search Application [54](#)
 - display name [50](#)
 - Encoding [51](#)
 - JSP tag libraries [54](#)
 - JSP tags [54](#)
 - Pattern [50](#)
 - viewing the search query page [53](#)
- Server Manager [13](#)
- Server-side HTML [67](#)
- Set Up a Client to Use WebDAV [45](#)
- Setting Up Database Connectivity [59](#)
- software virtual server. [18](#)
- Source URI [41](#)
- Specify a Directory Service for a Virtual Server [32](#)
- Start the Administration Server [12](#)
- Start the Web Server Instance [14](#)
- Sun Chili!Soft ASP [71](#)

U

URI [41](#)
 URL [11](#)
 User/Group page [37](#)
 User-based access control [27](#)
 Using ASP [71](#)
 Using CGI [67](#)
 Using Virtual Servers [17](#)

What are CGI programs? CGI
 What are CGI programs? [67](#)
 What is a web application? [55](#)
 What is access control? [27](#)

V

Virtual server [17](#)
 virtual server class [18](#)
 Virtual Server Manager [13, 22](#)

W

Web Applications
 running Java Web Applications On Your
 Server [55](#)
 sample applications [65](#)
 web container [55](#)
 web archive (WAR) file [55](#)
 web container [55, 59](#)
 Web Server Instance [11](#)
 Web-based Distributed Authoring and Versioning
 (WebDAV) protocol [39](#)
 WebDAV
 access control, restricting access [44](#)
 Create collection [43](#)
 Default search web application [49](#)
 Difference between URI and Source URI [41](#)
 Edit ACL [44](#)
 Enabling WebDAV globally [40](#)
 filesystem permissions [44](#)
 locking database [46](#)
 Set Up a Client to Use WebDAV [45](#)
 Set Up the Server to Handle WebDAV [39](#)
 Source URI [41](#)
 URI [41](#)

