Server Management Guide

Sun™ ONE Server Console

Version 5.2

816-6704-10
June 2003

Contents

About This Guide

Sun™ ONE Server Console software is part of the Sun Open Net Environment (Sun ONE), Sun's standards-based software vision, architecture, platform, and expertise for building and deploying Services On Demand.

Purpose of This Guide

This guide provides background information that system architects and administrators need to successfully install and manage Sun ONE servers in their enterprise. Read about Sun ONE server basics in this book before you begin installing and configuring Sun ONE servers.

Typographical Conventions

This section explains the typographical conventions used in this book.

`Monospaced font` - This typeface is used for literal text, such as the names of attributes and object classes when they appear in text. It is also used for URLs, filenames, and examples.

Italic font - This typeface is used for emphasis, for new terms, and for text that you must substitute for actual values, such as placeholders in path names.

The greater-than symbol (>) is used as a separator when naming an item in a menu or sub-menu. For example, Object > New > User means that you should select the User item in the New sub-menu of the Object menu.

NOTE	Notes, Cautions, and Tips highlight important conditions or limitations. Be sure to read this information before continuing.

Default Paths and Filenames

All path and filename examples in the Sun ONE Server Console product documentation are one of the following two forms:

> *ServerRoot/...* - The *ServerRoot* is the location of the Sun ONE Server Console product. The actual *ServerRoot* path depends on your platform, your installation, and your configuration. The default path depends on the product platform and packaging as shown in Table 1.

Table 1 Default *ServerRoot* Paths

Product Version	*ServerRoot* **Path**
Solaris Packages[1]	`/var/mps/serverroot` - After configuration, this directory contains links to the following locations: • `/etc/mps/admin/v5.2` (static configuration files) • `/usr/admserv/mps/admin` (Sun ONE Administration Server binaries) • `/usr/admserv/mps/console` (Server Console binaries)
Compressed Archive Installation on Solaris and Other UNIX Systems	`/var/Sun/mps`
Zip Installation on Windows Systems	`C:\Program Files\Sun\MPS`

[1] If you are working on the Solaris Operating Environment and are unsure which version of the Sun ONE Server Console software is installed, check for the existence a key package such as `SUNWasvu` using the `pkginfo` command. For example: `pkginfo | grep SUNWasvu`.

Suggested Reading

Useful information can be found on the following Web sites:

- Sun software: `http://wwws.sun.com/software/`
- Sun ONE Services: `http://www.sun.com/service/sunps/sunone/`
- Sun Support Services: `http://www.sun.com/service/support/`
- Sun ONE for Developers: `http://sunonedev.sun.com/`
- Training: `http://suned.sun.com/`

Part 1

Overview of Sun ONE Server Console

Chapter 1, “Sun ONE Server Console and Administration Server”

Chapter 2, “Installing Sun ONE Servers and Server Console”

Chapter 1

Sun ONE Server Console and Administration Server

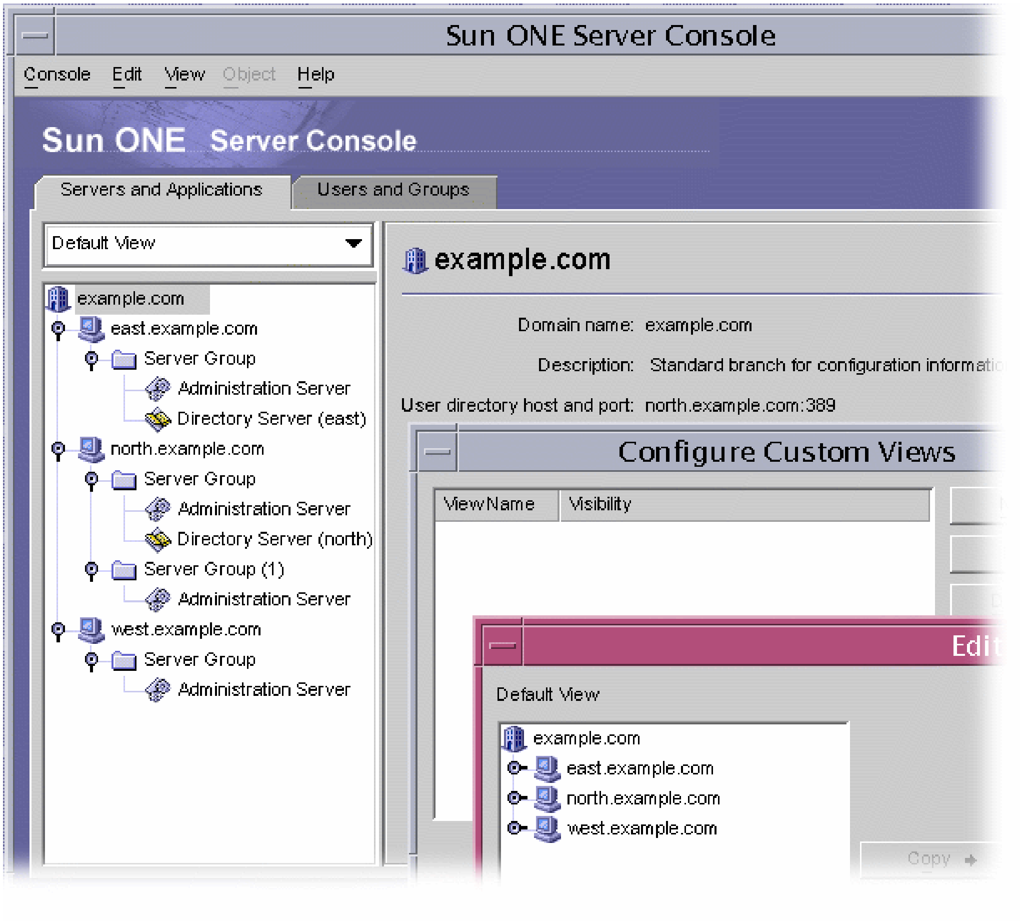
Sun ONE Server Console Version 5.2 and Administration Server Version 5.2 are two parts of a system that lets you manage Sun ONE server software in your enterprise. This chapter presents a high-level overview of what this system is and how you can use it to work with resources across your network.

Many Sun ONE servers depend on Sun ONE Directory Server. By default, when you do install Sun ONE Directory Server, Sun ONE Server Console and Administration Server are automatically installed for you. Although Sun ONE Directory Server, Sun ONE Server Console, and Sun ONE Administration Server work tightly with one another, each plays a specific role in the management of servers, applications, and users.

Sun ONE Directory Server stores server and application configuration settings as well as user information. This data is used by other servers in the enterprise. Typically, application and server configuration information is stored in one suffix of Sun ONE Directory Server while user and group entries are stored in another suffix. If you have a large enterprise, however, you can store your configuration and user information in separate *instances* of Directory Server (which can be on the same host machine or on two different host machines). When the terms *configuration directory* and *user directory* are used in this guide, they refer to where the configuration information and the user information is stored—either in the subtrees of a single instance of Directory Server or in two separate instances of Directory Server.

Sun ONE Server Console is the front-end management application for Sun ONE software in your enterprise. It finds all servers and applications registered in your configuration directory, displays them in a graphical interface, and lets you manage and configure them. In addition, Sun ONE Server Console provides graphical tools for locating and managing entries in the user directory. Figure 1-1 shows Sun ONE Server Console's interface.

Figure 1-1 The Sun ONE Server Console Interface



Domain name: example.com

Description: Standard branch for configuration informati

User directory host and port: north.example.com:389

When you log in to Sun ONE Server Console, it connects to an instance of Sun ONE Administration Server using the Hypertext Transfer Protocol (HTTP). Sun ONE Administration Server manages requests for all Sun ONE products installed in a single root folder.

When you install a Sun ONE product in a new folder, Sun ONE Administration Server is installed for you. If you install additional products in the same folder, they use the instance of Sun ONE Administration Server that is already there. If a product includes a newer version of Sun ONE Administration Server and Sun ONE Server Console than the versions in the root folder, the installer updates the folder with the latest versions.

The system for managing Sun ONE products works as follows:

Sun ONE Server Console lets you manage resources (servers or applications) as well as add or edit user information. When you use Sun ONE Server Console to manage resources, Console sends HTTP requests to the instance of Sun ONE Administration Server that controls the resource. Upon receiving these requests, the instance of Sun ONE Administration Server executes programs that perform the requested tasks. For example, Sun ONE Administration Server can execute programs to modify the server and application settings that are stored in the configuration directory or to change the port number that a server listens to.

When you use Sun ONE Server Console to add or edit user entries, it sends Lightweight Directory Access Protocol (LDAP) messages directly to Directory Server. The information in these messages is then stored in the user directory. Figure 1-2 illustrates the system.

Figure 1-2 Simple System With Sun ONE Server Console

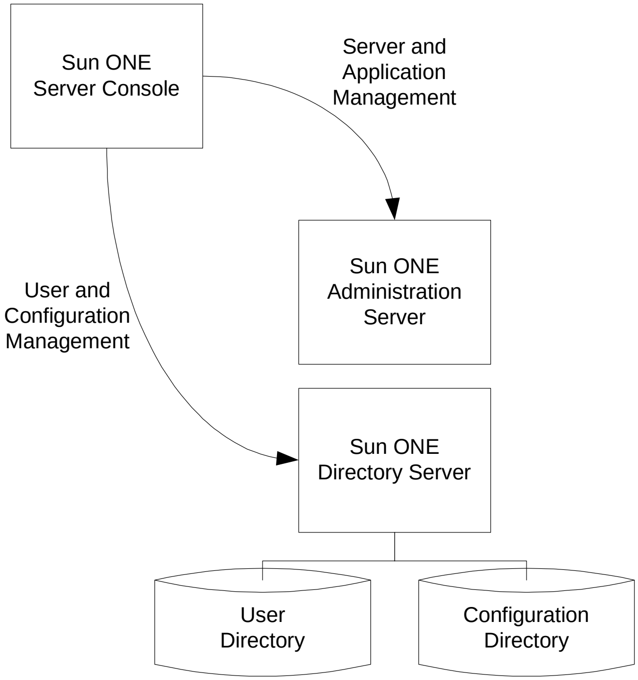
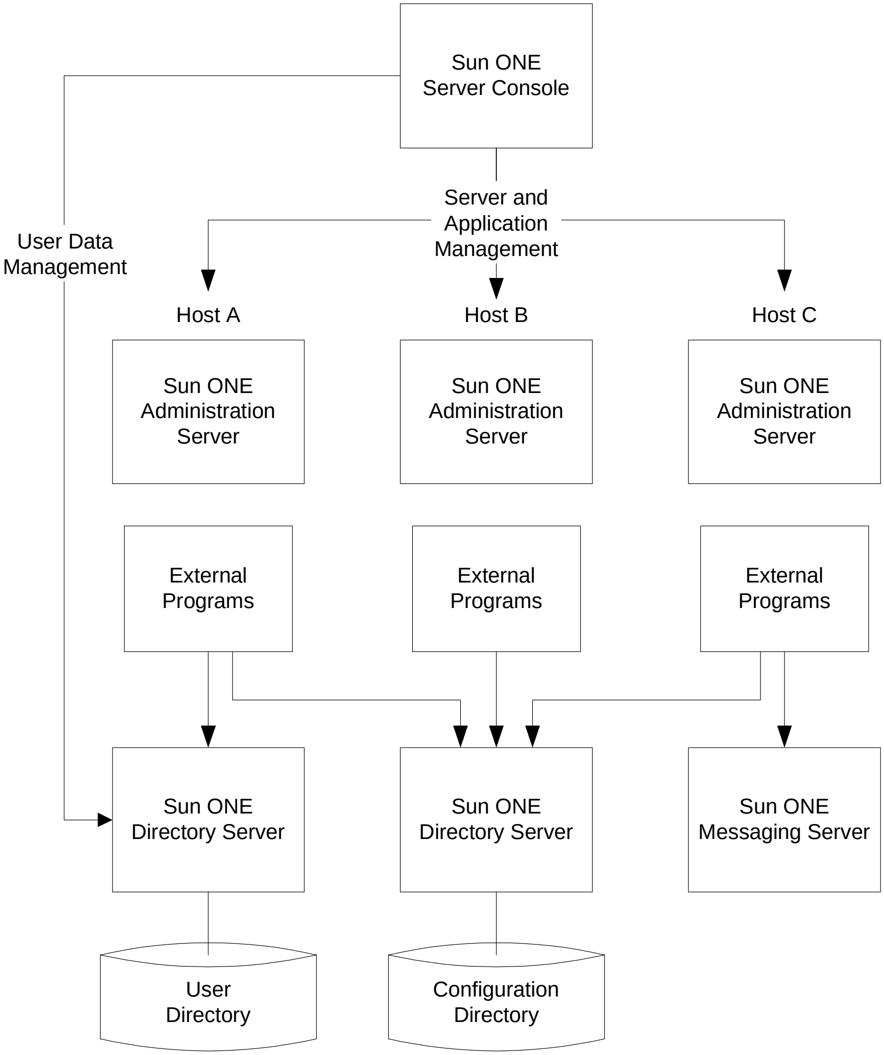

Sun ONE Server Console
Server and Application Management
Sun ONE Administration Server
User and Configuration Management
Sun ONE Directory Server
User Directory
Configuration Directory

Figure 1-2 shows an example of a relatively simple system. As your enterprise grows and your needs change, you have the flexibility to add additional hosts and servers. Even when you install new hardware and software, you can continue to use a single instance of Sun ONE Server Console to manage your network. Figure 1-3 shows how a complex system might be organized.

Figure 1-3 More Complex System With Sun ONE Server Console



The rest of this guide shows you how to install and use Sun ONE Server Console and Administration Server to manage servers, applications, and users.

If you would like to learn more about how Sun ONE Server Console works before installing the product, see "A Tour of Sun ONE Server Console," on page 33.

Chapter 2

Installing Sun ONE Servers and Server Console

This chapter provides an overview of the Sun ONE Server Products Setup program and how it is used in various situations.

This chapter contains the following sections:

- The Setup Program
- Upgrading to Version Version 5.2
- Silent Installation
- Uninstallation

Each Sun ONE server product has its own detailed installation instructions. To read these, see your server product documentation at `http://docs.sun.com`.

The Setup Program

The Sun ONE Server Products Setup program is for installing Sun ONE server products all at once or one at a time. Use the Setup program each time you need to do any of the following:

- Install a new server or server component
- Install Sun ONE Server Console as a stand-alone application
- Update a server

Installing a New Server

This section provides an overview of installation dependencies and options common to all Sun ONE server products.

TIP Each Sun ONE server has its own detailed installation instructions. Refer to the documentation for your server under `http://docs.sun.com/db/prod/sunone`

Directory Server Must Be Installed First

In order to install Sun ONE software, you must first set up Directory Server. When you do this, you create a user ID and password for the Configuration Administrator. During a typical installation, the Setup program checks this user ID and password against the installed directory. If the values do not match, authentication fails, and you can't complete the installation.

For detailed information on installing Directory Server, see the server's documentation at `http://docs.sun.com/db/prod/s1dirsrv`.

When you install Directory Server for the first time, Sun ONE Administration Server and Console are automatically installed for you.

Administration Server Is Required in Each Server Root

Every Sun ONE server root must contain an instance of Administration Server. If you are installing a server into a new folder, the Setup program automatically installs Administration Server for you.

NOTE Installing or upgrading Sun ONE Server Console on Windows requires a reboot at the end of the install process. If you choose not to reboot at the end of the install process you must remember to reboot later, before you use Sun ONE Server Console.

Administration Server Version 5.2 Is Not Backward Compatible

Due to changes in language support libraries, Sun ONE Administration Server Version 5.2 is not backward compatible. Therefore you must not install Sun ONE Administration Server Version 5.2 in the same server root as earlier Sun ONE server releases. As the Administration Server is required in each server root, this means you must install new servers in a new server root.

Interactive Installation Modes

Server installation programs offer three interactive installation modes: Express, Typical, and Custom.

Express

Use this mode to get the system running quickly, using default settings as much as possible. This mode was designed for administrators who want to test a server's basic operation on a particular system before deploying. It automatically generates as much information as possible to complete the most basic installation. Generally, you need to enter only administrator names and passwords during an express installation.

Typical

Use this mode if you want to specify some, but not all, installation options. Administrators often use this mode because it handles the details of server configuration, while still letting administrators modify settings such as directory location, port numbers, user names, and passwords.

Custom

Use this mode only if you have run the installer before, and are familiar with server configuration settings and how to modify them. This mode is most useful to the administrator who routinely installs and upgrades servers, and whose company has already identified special enterprise needs. When using custom mode, you can specify every typical option as well as advanced ones such as the IP address of a host system.

Installing Sun ONE Server Console as a Stand-Alone Application

You can install Sun ONE Server Console as a stand-alone application on a local workstation. Having Sun ONE Server Console on your local system allows you to manage servers on remote hosts.

To Install Solaris™ Packages

When using the product delivered in Solaris package format, install the packages listed in Table 2-1 using the pkgadd(1M) utility before setting up the Sun ONE Server Console.

Table 2-1 Sun ONE Server Console Required Packages

Package	Description
`SUNWasvc`	Sun ONE Server Console
`SUNWasvcp`	Sun ONE Administration Server Console Plug-In
`SUNWasvr`	Sun ONE Administration Server (Root)
`SUNWasvu`	Sun ONE Administration Server (Usr)

To Setup the Sun ONE Server Console

1. When using the Solaris packaged version, enter `/usr/sbin/mpsadmconfig configure`.

   When not using the Solaris packaged version, change to the location where you unpacked the server product software, and then run the setup program for your server.

   The first installation or configuration screen appears.

2. Follow the instructions on screen, opting to install only Sun ONE Server Console.

   The Setup program installs Sun ONE Server Console in the location you specify.

After installation is complete, you can run the Console on:

- Solaris systems on which the packaged version of the software is installed by entering `/usr/sbin/mpsconsole startconsole`.
- UNIX® systems by changing to the directory where you installed Sun ONE Server Console, and then entering `./startconsole`.
- Windows systems by clicking Start, and then choosing Programs > Sun ONE Server Products > Sun ONE Server Console Version 5.2.

Upgrading to Version Version 5.2

If you already have earlier versions of the product installed on your system, you can upgrade to Sun ONE Server Console Version 5.2. This section contains instructions for performing the following upgrades:

- Upgrading Administration Server and Console
- Upgrading a Stand-Alone Console.

NOTE The instructions presented in this section apply only when upgrading the Administration Server and Server Console. If you want to upgrade a different Sun ONE product, refer to the installation instructions for the upgraded version of that product under `http://docs.sun.com/db/prod/sunone`

Upgrading Administration Server

As described on page 18, "Administration Server Version 5.2 Is Not Backward Compatible." Upgrading to Sun ONE Administration Server Version 5.2, involves installing the Administration Server alongside a new Sun ONE server in a new server root.

Upgrading a Stand-Alone Version of Sun ONE Server Console

If you have installed a stand-alone version of Sun ONE Server Console, you may upgrade to version Version 5.2 using the following procedures.

To Install Required Solaris Packages

Before setting up the Sun ONE Server Console, install the packages listed in Table 2-1 on page 20 using the `pkgadd`(1M) utility.

To Setup the Sun ONE Server Console

1. When using the Solaris packaged version, enter `/usr/sbin/mpsadmconfig configure`.

When not using the Solaris packaged version, change to the location where you unpacked the server product software, and then run the setup program for your server.

The first installation screen appears.

2. Follow the instructions on screen, opting to install only Sun ONE Server Console.

3. When not using the Solaris packaged version, ensure that you install the Sun ONE Server Console in the same location as the existing version.

After the upgrade is complete, you can run the Console on:

- Solaris systems on which the packaged version of the software is installed by entering `/usr/sbin/mpsconsole startconsole`.
- UNIX® systems by changing to the directory where you installed Sun ONE Server Console, and then entering `./startconsole`.
- Windows systems by clicking Start, and then choosing Programs > Sun ONE Server Products > Sun ONE Server Console Version 5.2.

Silent Installation

The Silent Installation feature of the Sun ONE Server Products Setup program allows you to use a file to predefine all the specifications that you would normally supply interactively during installation of each server. Silent Installation is useful when you want to install a large number of Sun ONE server instances using identical installation options.

Performing a Silent Installation

In order to perform a silent installation, you must create a set of installation specifications and then run the Sun ONE Server Products Setup program in silent mode. The easiest way to create a set of installation answers is to perform an installation and save your installation cache to a file. Once you have done this, you can modify the file and then use it when performing additional installations.

You can use Silent Installation to upgrade multiple instances of Administration Server. Rather than entering the same set of answers for each server manually, you save your installation answers while upgrading one server instance, and then upgrade the remaining instances using similar answers.

For detailed information on how a particular server uses Silent Installation, refer to the installation documentation for your server.

Uninstallation

If you are no longer using an Sun ONE server, you may uninstall it. Uninstallation completely removes a server and accompanying data from your computer. The server is no longer accessible and you lose all settings.

For detailed information on how to uninstall a particular server, refer to the documentation for your server.

Part 2

Sun ONE Server Console Basics

Chapter 3

Using Sun ONE Server Console

This chapter shows you how to log in to, customize, and use Sun ONE Server Console. It contains the following sections:

- Starting Sun ONE Server Console and Logging In
- A Tour of Sun ONE Server Console
- Customizing Sun ONE Server Console

Starting Sun ONE Server Console and Logging In

Sun ONE Server Console is a stand-alone Java application that works in conjunction with an instance of Directory Server and an instance of Administration Server on your network. Typically, you log in to Sun ONE Server Console using your own user name and password. If the instance of Administration Server that you are logging in to requires client authentication, you are prompted to present a client certificate. This certificate is used to create a secure channel of communication between Sun ONE Server Console and the instance of Administration Server. This section describes login procedures for both cases.

Starting Sun ONE Server Console

The following procedures explain how to start Sun ONE Server Console on supported platforms.

To Start Sun ONE Server Console on UNIX Systems

When using the Solaris packaged version, enter *basedir*`/usr/sbin/mpsadmserver startconsole` *[arguments]* where basedir is the base directory where you installed the packages (by default package contents are placed under `/`), and *arguments* are any of the optional command-line arguments listed in Table 3-1.

When not using the Solaris packaged version, change to the server root directory, and then enter `startconsole` *[arguments]* where *arguments* are any of the optional command-line arguments listed in Table 3-1.

To Start Sun ONE Server Console on Windows Systems

Click Start, and then choose Programs > Sun ONE Server Products > Sun ONE Server Console Version 5.2.

Alternatively, you can start Sun ONE Server Console in two additional ways:

- Double-click `startconsole` in your server root.
- Enter `startconsole` *[arguments]* at the command prompt where *arguments* are any of the optional command-line arguments listed in Table 3-1.

Table 3-1 `startconsole` Command Line Arguments

Argument	What it Does
`-a` *adminURL*	Specifies a base URL for the instance of Administration Server to use such as `http://server.example.com:1389`.
`-f` *fileName*	Captures errors and system messages to *fileName*.
`-h`	Displays a usage message explaining command line options.
`-l` *languageCode*	Specifies which language this version of Sun ONE Server Console should use. Supported values for *languageCode* are `en` (English), `fr` (French), and `ja` (Japanese).
`-u` *userID*	Specifies the user logging in to Sun ONE Server Console.
`-w` *password*	Specifies the password for the user entered with the `-u` argument.
-x *extraOptions*	Specifies that you want to use extra options. Supported values for *extraOptions* are `nowinpos` and `nologo`. If you specify the `nologo` option, the Sun ONE Server Console splash screen is not displayed. If you specify the `nowinpos` option, the Sun ONE Server Console window is placed in the upper left corner of the screen. To specify both options, separate them with a comma.

Logging In to Sun ONE Server Console With a User Name and Password

The following procedure explains how to log in to Sun ONE Server Console with a user name and password only. If you are logging in to an instance of Administration Server that requires you to present a client certificate, see "Logging In to Sun ONE Server Console Using Client Authentication," on page 30.

To Log in to Sun ONE Server Console With a User Name and Password

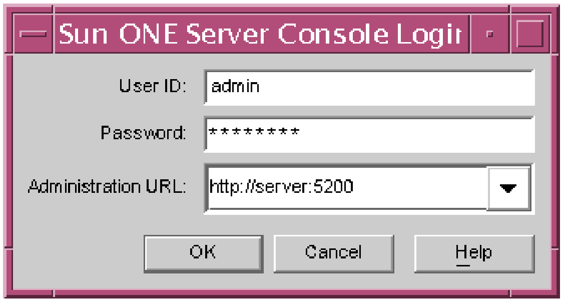
1. Start Sun ONE Server Console.

   For more information, see "To Start Sun ONE Server Console on UNIX Systems," on page 28 or "To Start Sun ONE Server Console on Windows Systems," on page 28.

2. In the Sun ONE Server Console Login dialog box, enter your user name, password, and the URL for the instance of Administration Server you want to access.

   When specifying an Administration Server URL, you can use a host name and port number or IP address and port number. You do not need to include `http://` or use a fully qualified domain name, but you must include the Administration Server port number.

Figure 3-1 Sun ONE Server Console Login


Sun ONE Server Console Logir

User ID: admin

Password: ********

Administration URL: http://server:5200

3. Click OK.

   The user name and password you use to log in determine which servers and server operations you can access through Sun ONE Server Console. See "Overview of Access Control" on page 129 for more information.

TIP Sun ONE Server Console remembers the last five Administration URLs entered. To use one of these URLs, select it from the drop-down list in the Administration URL field.

Logging In to Sun ONE Server Console Using Client Authentication

When logging in to an instance of Administration Server that has been configured to require client authentication, you enter your user name and password, and then present a client certificate. This certificate is used by the instance of Administration Server to establish an SSL-enabled connection with Sun ONE Server Console. For more information on this process, known as the Secure Sockets Layer (SSL) handshake, see Appendix B, "Introduction to SSL."

The client certificates that Sun ONE Server Console presents to an instance of Administration Server are stored in Netscape Communicator certificate database format. New and existing certificates are not recognized by Administration Server unless they are stored in the Netscape Navigator 4.7x certificate database format. For initial setup of client authentication, store certificates in the Netscape Navigator browser. After initial setup certificates can be stored in other browser certificate databases. For more information about Netscape Navigator certificate database format and certificate storage see "To Set Up Client Authentication for Users" on page 169.

Depending on which types of certificates the instance of Administration Server is configured to accept, you may be able to use an existing certificate, or you may need to request a new one. You may use Netscape Communicator to request and install client certificates.

This section tells you how to do the following:

- Request and install a new client certificate
- Make your client certificate available to Sun ONE Server Console
- Establish a secure connection with an instance of Administration Server

For more information on configuring an instance of Administration Server to require client authentication, see Chapter 10, "Using SSL and TLS with Sun ONE Servers."

To Request and Install a New Client Certificate

1. Go to the web site for a certificate authority (CA) that is trusted by the instance of Administration Server that you want to establish a secure connection with.

2. Follow the CA's instructions to request and install a client certificate.

> **NOTE** If you already have a client certificate that is acceptable to the instance of Administration Server you use, you do not need to request and install a new certificate.

To Make Your Client Certificate Available to Sun ONE Server Console on UNIX Systems

1. From the system prompt, go to the `.netscape` subdirectory of your home directory. For example, `/home/bjensen/.netscape`.

2. Copy the `key3.db`, `cert7.db`, and `secmod.db` files to the `.mcc` subdirectory of your home directory.

   These files are the certificate database files that Sun ONE Server Console uses during client authentication. These files are only used by Sun ONE Server Console. Administration Server creates and uses its own certificate database files.

To Make Your Client Certificate Available to Sun ONE Server Console on Windows

1. Open the folder containing Netscape Communicator. For example, `C:\Program Files\Netscape`.

2. Open the `Users` folder and then open your specific user folder. For example, `BJensen` (`C:\Program Files\Netscape\Users\BJensen`).

3. Copy the `key3.db`, `cert7.db`, and `secmod.db` files from your user folder to the `C:\WINNT\Profiles\`*your_user_ID*`\.mcc` folder, where *your_user_ID* is the ID that you use to log in to Windows.

   These files are the certificate database files that Sun ONE Server Console uses during client authentication. These files are only used by Sun ONE Server Console. Administration Server creates and uses its own certificate database files.

To Establish a Secure Connection With an Instance of Administration Server

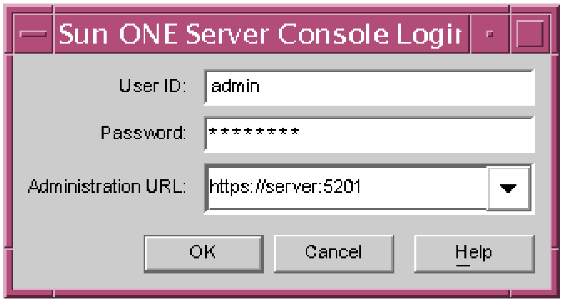
1. Start Sun ONE Server Console.

   For more information, see "To Start Sun ONE Server Console on UNIX Systems," on page 28 and "To Start Sun ONE Server Console on Windows Systems," on page 28.

2. In the Sun ONE Server Console Login dialog box, enter your user name, password, and the URL for the secure instance of Administration Server you want to access.

   When specifying an Administration Server URL, you can use a host name and port number or IP address and port number. Make sure to include `https://` and the Administration Server port number in the URL.

Figure 3-2 Secure Sun ONE Server Console Login



User ID: admin

Password: ********

Administration URL: https://server:5201

3. Click OK.

   The user name and password you use to log in determine which servers and server operations you can access through Sun ONE Server Console. See "Overview of Access Control" on page 129 for more information.

4. In the Password Entry dialog box, enter the password for the Sun ONE Server Console certificate database (this is the same as the password for your Netscape Communicator certificate database), and then click OK.

5. In the “Select a Certificate” dialog box, select your client certificate from the drop-down list, and then click OK.

   Sun ONE Server Console presents this certificate to the instance of Administration Server. If the instance of Administration Server is configured to accept certificates from your CA, your user name and password are authenticated, and you see the Sun ONE Server Console interface. Otherwise, you are prompted to select a different certificate.

A Tour of Sun ONE Server Console

After you log in to an Administration Server, you see the Sun ONE Server Console interface. This section introduces the graphical elements of this interface and explains the basic concepts you need to understand before managing Sun ONE servers with Sun ONE Server Console.

Sun ONE Server Console Menus

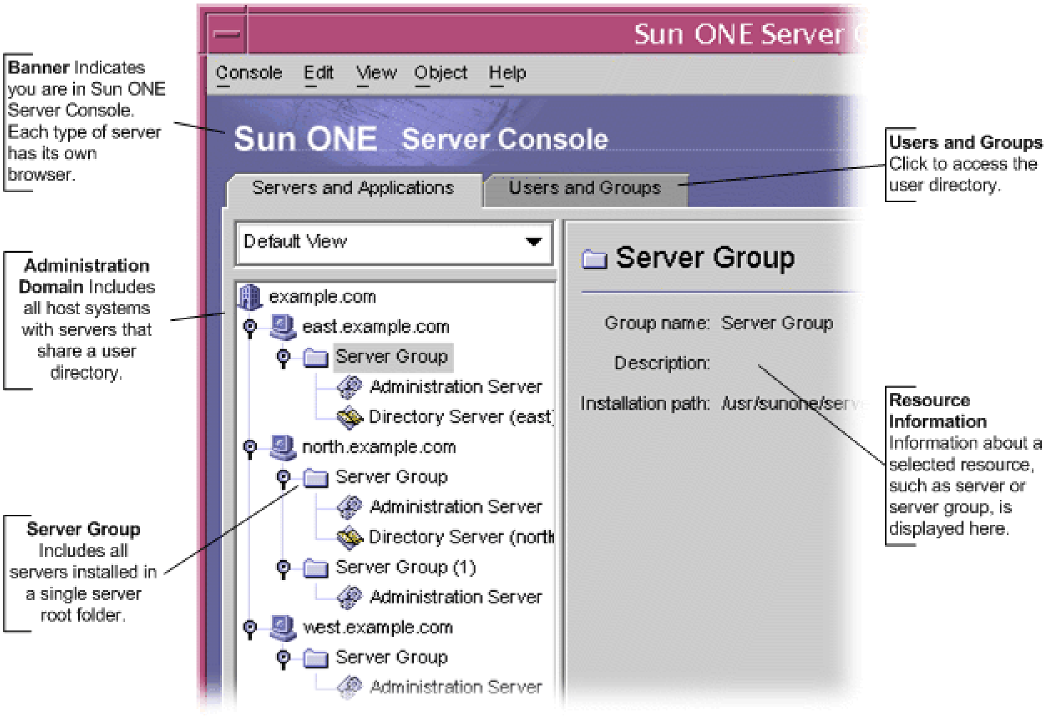
The main Sun ONE Server Console window shown in Figure 3-3 on page 34 has five menus: Console, Edit, View, Object, and Help. Table 3-2 summarizes what these menus are used for.

Table 3-2 Sun ONE Server Console’s Menus and What You Can Do With Them

Menu	What It Lets You Do
Console	Add and remove items from the navigation tree.
Edit	Set general Sun ONE Server Console preferences.
View	Change the appearance of the main Sun ONE Server Console window.
Object	Perform tasks related to resources such as administration domains, server groups, and servers.
Help	Obtain online assistance while using Sun ONE Server Console.

Other Sun ONE products may have additional menus or use these menus differently. For more information, see the documentation for each product.

Figure 3-3 Main Sun ONE Server Console Window



Group name: Server Group

Description:

Installation path: /usr/sunone/serv

Banner Indicates you are in Sun ONE Server Console. Each type of server has its own browser.

Administration Domain Includes all host systems with servers that share a user directory.

Server Group Includes all servers installed in a single server root folder.

Users and Groups Click to access the user directory.

Resource Information Information about a selected resource, such as server or server group, is displayed here.

Sun ONE Server Console Tabs

The main Sun ONE Server Console window (shown in Figure 3-3) has two tabs: "Servers and Applications" and "Users and Groups." The "Servers and Applications" tab contains a navigation tree and an information panel. The "Users and Groups" tab has an interface that you can use to manage entries in the user directory. The "Users and Groups" tab is discussed in Chapter 5, "User and Group Administration."

The Servers and Applications Tab

The “Servers and Applications” tab consists of a navigation tree and an information panel. The navigation tree represents a Sun ONE *topology*. A topology is a hierarchical representation of all the *resources*, or objects (such as servers, applications, and hosts), that are registered in a configuration directory. You use the navigation tree to navigate to the resource you want to work with.

One type of resource in a topology is an *administration domain*. An administration domain is a collection of host systems and servers that share a user directory.

A number of *server groups* can exist within an administration domain. A server group consists of one or more servers that are managed by a common instance of Administration Server and that share a server root folder. The individual *servers* in a server group are instances of server software that provide specific services such as directory database services, messaging, and publishing.

Figure 3-3 shows a sample navigation tree. In this example, the `example.com` administration domain includes three hosts. If the administration domain grows, an administrator can install additional server groups on these hosts.

On the right-hand side of the “Servers and Applications” tab is the *information panel*. When you select an administration domain, host, server group, or server instance in the navigation tree, this panel displays detailed information about it. Depending on the selected resource, you can edit all or some of these details.

For information on modifying administration domain settings, see “To Modify an Administration Domain,” on page 37. For information on modifying host, server group, and instance information, see “Modifying Host, Server Group, and Instance Information,” on page 49.

The Administration Domain

An administration domain is a group of Sun ONE server products that share a user directory for data management and authentication. A company might want to create separate administration domains for each of its business sites. Each of these domains could include the host computers used only by that business site.

Before you can create a new administration domain, you must be a member of the Configuration Administrators group. If you are not a member of this group, you must ask your Configuration Administrator to add you to it. For instructions on adding a user to the Configuration Administrators group, see “To Add Users to the Configuration Administrators Group,” on page 77.

To Create an Administration Domain

1. Open Sun ONE Server Console.

2. From the Console menu, choose Create Administration Domain.

3. In the Create Administration Domain dialog box, enter domain information:

   Domain Name. Enter a name that helps you identify this domain. This can be a fully qualified domain name such as `example.com` or a descriptive title such as East Coast Sales.

   User Directory Host. Specify the host machine on which the user directory for this domain is located. Use the fully qualified domain name. For example, `east.example.com`.

   User Directory Port. Enter the port number for the user directory you specified above.

   Secure Connection. Check this box if you want to connect to the user directory using SSL. If you select this option, make sure that the user directory port you've entered is already enabled for SSL communication.

   Directory Subtree. Enter the base DN of the user subtree in the directory. Example: `dc=example,dc=com`

   Bind DN. Enter the distinguished name for a user who has full access permission to the user directory. Example: `uid=jdoe, ou=people, dc=example,dc=com`.

   Bind Password. Enter the password for the user specified by the Bind DN.

   Owner DN. Enter the distinguished name for the user who has administrative control over this domain. By default, your DN is entered.

4. Click OK.

   If you've made a change to the User Directory option or the Secure Connection option, you must restart the server for the change to take effect.

To Modify an Administration Domain

1. In the Sun ONE Server Console navigation tree, select the domain you want to modify, then click the Edit button in the server information panel of Sun ONE Server Console.

2. Modify domain information as necessary:

   Domain Name. Enter the name of the domain as you want it to appear in the navigation tree.

   Description (Optional). Enter a text string that helps you identify this domain.

   User Directory Host and Port. Specify the location of the user directory using the host computer's fully qualified domain name and port number. You can enter more than one user directory location separated by spaces. This is useful when you use multiple directories to allow users to log in if a primary Directory Server is inaccessible. Example:

   ```
   east.example.com:389 west.example.com:389
   ```

   See "User Authentication and Directory Failover Support," on page 108 for more information.

   All host computers specified in the User Directory Host and Port field must have the same settings for the following fields:

   Secure Connection. Check this box if the new user directory port is already enabled for SSL communication.

   User Directory Subtree. Enter the base DN of the user information in the new user directory. Example: `dc=example,dc=com`

   Bind DN. Enter the distinguished name for a user who has full access permission to the new user directory. Example: `uid=jdoe, ou=people, dc=example,dc=com`.

   Bind Password. Enter the password for the user specified by the Bind DN.

CAUTION These settings affect all servers in the domain. If you make changes here, you must restart all servers in the domain.

3. Click OK.

To Remove an Administration Domain

1. Open Sun ONE Server Console.
2. Remove all server instances from the administration domain that you want to remove.

   For more information on removing server instances, see “Removing a Server Instance,” on page 51.
3. Select the administration domain that you want to remove.
4. From the Console menu, choose Remove Administration Domain.
5. Click OK.

Customizing Sun ONE Server Console

This section tells you how to specify where to store display settings as well as how to change Sun ONE Server Console’s appearance to meet your specific needs. It explains the following:

- How to specify where Sun ONE Server Console should store your display preferences.
- How to specify which fonts Sun ONE Server Console should use for on screen elements.
- How to change the width and position of columns in tables.
- How to customize views of the navigation tree.

In addition, you can change Sun ONE Server Console’s appearance by applying access control instructions to user interface elements. This procedure is discussed in Chapter 9, “Access Control.”

Storing Display Settings

When you exit Sun ONE Server Console, any display changes you’ve made during the session are saved. This includes changes to window size or position; banner bar, status bar, or navigation tree visibility; and fonts.

You can store these display settings on the network or on your local disk to suit your needs. If, at any time, you want the settings reset to what they were when you installed Sun ONE Server Console, you can do so.

To Change Where Display Settings Are Stored

1. In Sun ONE Server Console, from the Edit menu, choose Preferences.

2. Click the Settings tab.

3. Specify where you want to save your display settings:

   In your configuration directory. Select this option if you want to be able to use your settings no matter where you are when you log in to Sun ONE Server Console. This option is useful if you frequently “roam” between a number of similar workstations at your business site. No matter what workstation you’re using, when you log in to Sun ONE Server Console you can use your preset display preferences.

   On your computer’s hard disk. Select this option if you want to be able to use different display settings depending upon the individual workstation you’re using. This option is useful when you use one workstation at work and a dissimilar system, such as a laptop computer, at home. The settings for the workstation are stored and used on the workstation. The settings for the laptop are stored and used on the laptop.

4. Click OK.

To Reset Display Settings to Their Default Values

1. In Sun ONE Server Console, from the Edit menu, choose Preferences.

2. Click the Settings tab.

3. Click the Restore Defaults button to revert to the default display settings.

4. Click OK.

Setting Display Fonts

You can specify which fonts Sun ONE Server Console should use for different screen elements. If you use more than one computer system to administer servers, you can save different sets of font preferences, or *profiles,* for use on each system.

To Create a Font Profile

1. In the main Sun ONE Server Console window, from the Edit menu, choose Preferences.

2. Click the Fonts tab.

3. Click Save As, enter a name for this profile, and then click OK.
4. In the Screen Element column, click a screen element that you want to change the font for.

   The Font column contains samples of the fonts that are currently associated with the listed screen elements.
5. Click Change Font.

   The Select Font dialog box appears.
6. In the Select Font dialog box, make your font selections:

   Font. Choose the font face you want to use for this element.

   Size. Choose a size for the selected font face.

   Bold. Select this option to display the font in bold.

   Italic. Select this option to display the font in italics.

   Sample. This frame displays sample type using the current settings.
7. Click OK to close the Select Font dialog box.
8. If you want to set fonts for additional screen elements, repeat steps 4 through 7.
9. Click OK to save the profile.

To Edit an Existing Font Profile

1. In the main Sun ONE Server Console window, from the Edit menu, choose Preferences.
2. Click the Fonts tab.
3. Select the font profile to edit.

   From the Font Profile drop-down list, choose a profile. If the list is grayed out, no profiles are available.
4. Make the desired changes to the font profile.
5. Click OK to save the profile.

To Rename a Font Profile

1. In the main Sun ONE Server Console window, from the Edit menu, choose Preferences.
2. Click the Fonts tab.
3. Select the font profile to rename.

   From the Font Profile drop-down list, choose a profile. If the list is grayed out, no profiles are available.
4. Click Save As, enter the new name for this profile, and then click OK.

   A new profile with the name you specified appears in the Font Profile drop-down list. The original profile is still listed.
5. From the Font Profile drop-down list, select the original font profile.
6. Click Remove, and then confirm the deletion.
7. Click OK to save the renamed profile.

Customizing the Main Window

You can specify which elements of the main Sun ONE Server Console window you want to see.

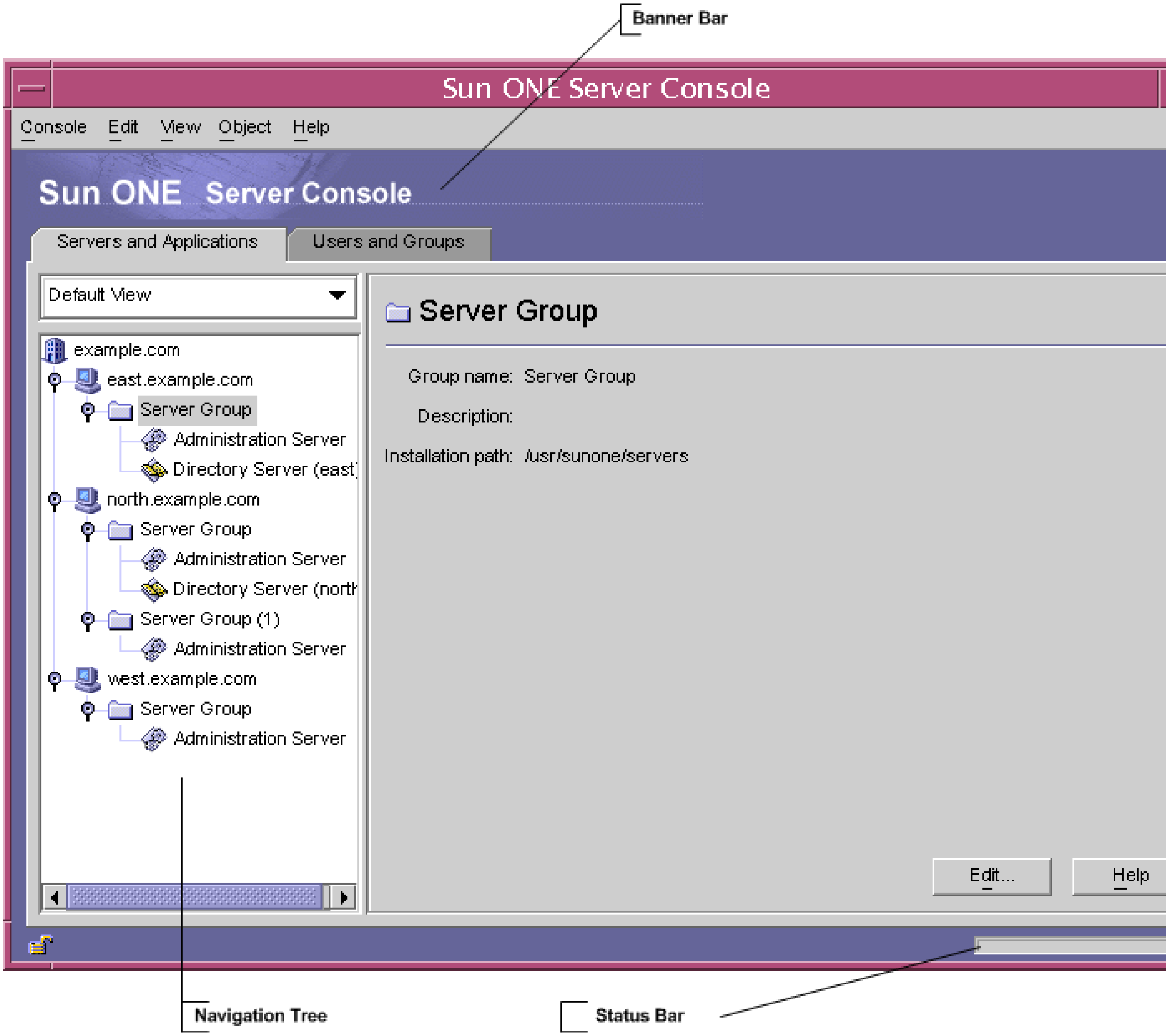
To Customize the Main Window

Select or deselect items in the View menu.

Selecting a menu item displays it and deselecting an item hides it. You can show or hide the following screen elements:

- Banner Bar
- Status Bar
- Navigation Tree

Figure 3-4 Banner, Navigation Tree, and Status



Group name: Server Group

Description:

Installation path: /usr/sunone/servers

Banner Bar

Navigation Tree

Status Bar

Creating Custom Views of the Navigation Tree

You can create custom views of the navigation tree. Custom views are useful when you want to see the resources that you access routinely, and hide resources that you access infrequently.

When creating a custom view, you can specify whether the view is public or private. A public view is visible to any user who logs in to Sun ONE Server Console. A private view is visible only to the person who created it.

To Create a Custom View of the Navigation Tree

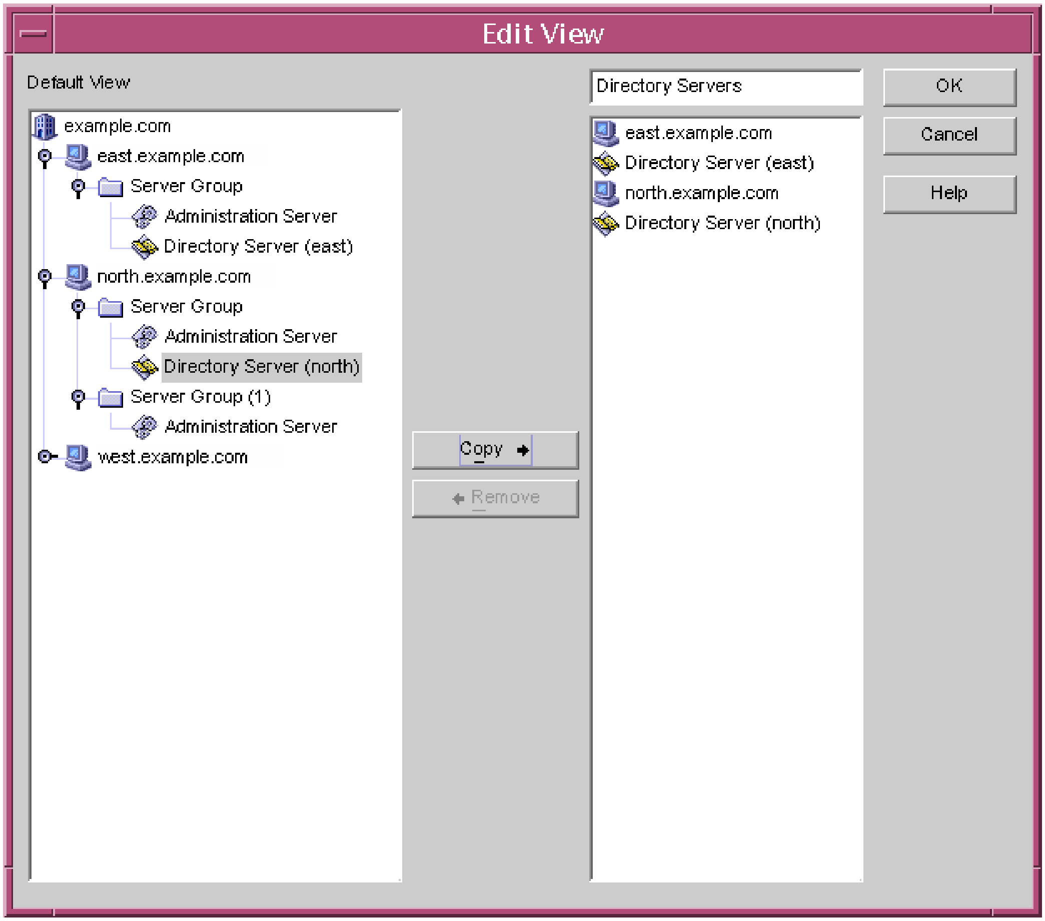
1. From the View menu, choose Custom View Configuration, then click New.
2. Choose whether the new view is public or private, then click OK.

   By default, a public view is visible to all users of Sun ONE Server Console, but you can restrict access to it using access control instructions (ACIs). For more information, see “To Set Access Permissions for a Public View,” on page 46.

   A private view is only visible to you. You cannot apply ACIs to it.
3. Use the Edit View window shown in Figure 3-5 on page 44 to customize the tree.
4. Click OK when you have finished adding resources.

In the example that follows, an administrator has created a view named Directory Servers that includes instances of Sun ONE Directory Server and their hosts.

Figure 3-5 Customized Navigation Tree



Directory Servers

Working With Custom Views

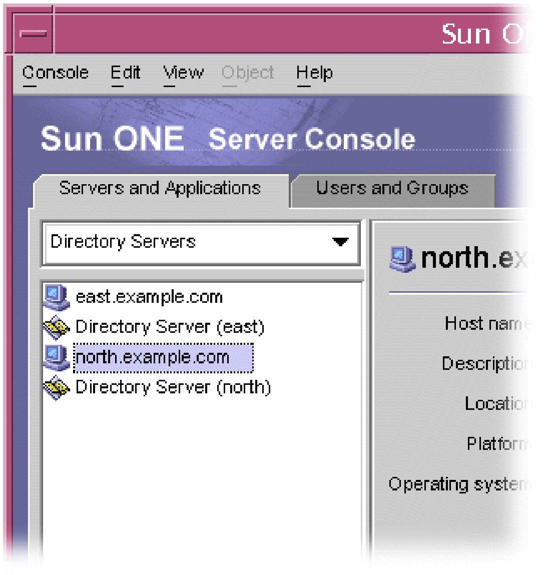
You can use multiple views to suit your needs. The administrator who created the view shown in the preceding example might also have views called Directory Servers and Enterprise Servers. The administrator can switch to the Custom View needed for a specific task or choose Default View to see all the servers in the navigation tree.

When you install Sun ONE Server Console, a Custom View called Server View is configured for you. This view displays server instances grouped by type; it does not include administration domains, hosts, or server groups.

To Switch to a Custom View

Choose the desired custom view from the drop-down list on the "Servers and Applications" tab. To return to the default view, choose Default View from the drop-down list.

Figure 3-6 Switching to a Custom View



To Edit a Custom View

1. From the View menu, choose Custom View Configuration.
2. Select a Custom View from the list and click Edit.
3. Make any necessary changes to the Custom View.
4. Click OK.

To Rename a Custom View

1. From the View menu, choose Custom View Configuration.
2. Choose a Custom View from the list and click Edit.
3. In the Edit View window, position the cursor in the text field, then type the new name for your Custom View.
4. Click OK.

To Set Access Permissions for a Public View

1. From the View menu, choose Custom View Configuration.
2. Choose a public Custom View from the list and click Access.
3. Specify the ACI you want to use, or create a new ACI:
    - If you want to use an existing Access Control Instruction (ACI), select it and click OK.
    - If you want to create a new ACI, click New, and then follow the directions for creating a new ACI under “Using the ACI Manager and ACI Editor.” beginning on page 134.
4. Click OK when you have finished setting access permissions.

For more information on setting Access Permissions and creating Access Control Instructions, see Chapter 9, “Access Control.”

To Delete a Custom View

1. From the View menu, choose Custom View Configuration.
2. Choose a Custom View from the list and click Delete.
3. Click Yes to confirm the deletion.

Chapter 4

Servers in Sun ONE Server Console

This chapter explains how to perform basic server management using Sun ONE Server Console. You can perform a number of basic server tasks with Sun ONE Server Console. This section contains the following procedures:

- Opening a Server Management Window
- Creating a New Server Instance
- Modifying Host, Server Group, and Instance Information
- Cloning a Server
- Removing a Server Instance
- Uninstalling a Sun ONE Server
- Merging Configuration Data From Two Directory Servers

Opening a Server Management Window

Each Sun ONE server has its own set of tasks and configuration settings. You can access these by opening a server management window.

To Open a Sun ONE Server Management Window

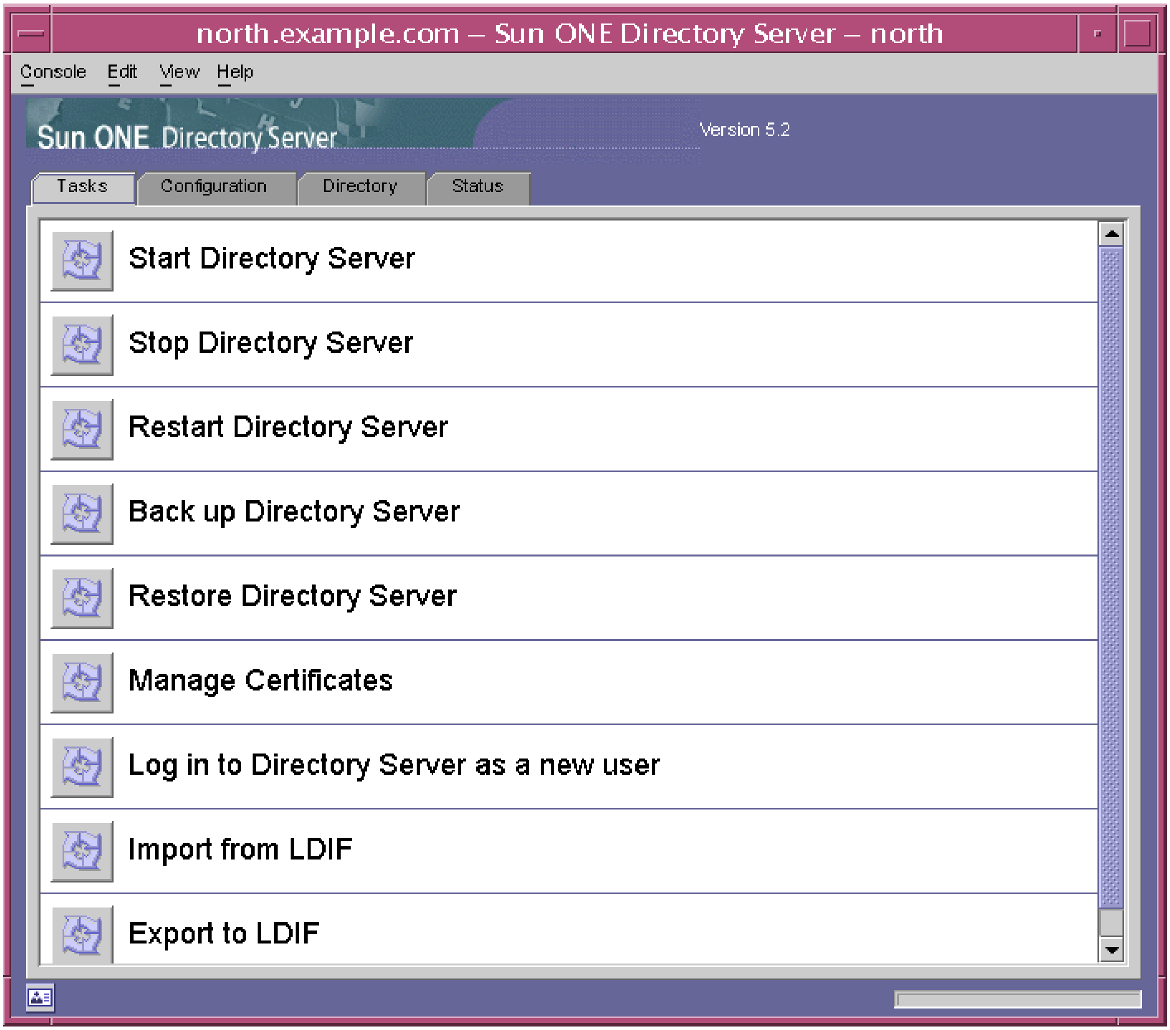
1. In Sun ONE Server Console, click the “Servers and Applications” tab to see the navigation tree on the left and server information on the right.
2. In the navigation tree, click a server to select it.
3. In the information panel on the right side of the window, click Open.

Another way to open a server management window is by double-clicking its icon in the navigation tree.

Each Sun ONE server has specialized tabs for such functions as setting configurations and viewing server-specific information. For detailed information about a specific tab, see your server's documentation. You can view many of the guides for specific products online at

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

Figure 4-1 Server Management Window



Creating a New Server Instance

Once you have one instance of a server installed in a server root, you can create additional instances in the same server root. Having multiple instances in a single server root is useful for testing and for when one host is used for multiple purposes.

For example, a company's Human Resources and Finance departments each need a web server. As each department has limited publishing requirements, one host can serve both departments' needs. The administrator installs the web server software once, creating one instance of the server, and then creates a second instance. One instance is for the Human Resources department and the other is for the Finance department. Only one instance can run on the default web server port (80); the administrator must assign a different port number to the other instance.

NOTE You cannot create two instances of Administration Server in the same server root.

To Create a New Server Instance

1. In Sun ONE Server Console, select the server group to contain the new server instance.
2. From the Object menu, select Create Instance Of.
3. In the Select Server submenu, select the server that you want to create a new instance of.

   A dialog box is displayed, allowing you to provide basic configuration information for the new server instance. This dialog box is specific to the server you are creating.
4. After providing the appropriate configuration information, click OK.

Modifying Host, Server Group, and Instance Information

You can edit some of the host, server group, and instance information that Sun ONE Server Console displays in the information panel. This is useful when you want to add detailed descriptions of the different installations in your organization.

To Modify Host, Server Group, and Instance Information

1. In the Sun ONE Server Console navigation tree, select the host, server group, or instance for which you want to modify information.

2. In the information panel, click Edit.

3. Edit information for the following fields:

   Host/Group/Server Name. Enter a descriptive name for this host, server group, or instance. Examples:

   - `Midwest Server`
   - `East Coast Sales Servers`
   - `West Coast Messaging Server No. 3 (P-Z)`.

   Description. Enter a detailed description of this server group or instance. Examples:

   - `Midwestern team's primary directory and messaging server.`
   - `The server group containing the East Coast Sales team's instances of Messaging Server and Certificate Management System`
   - `The West Coast Messaging Server for users with last names beginning with P through Z.`

   Location. (Host only) Enter a description of this host's location. Example: `Building 17, 3rd floor, Lab 1749`.

4. Click OK.

Cloning a Server

Cloning allows you to copy one server's configuration settings to other servers of the same type.

NOTE This operation is not supported for all Sun ONE servers. Before proceeding, refer to the documentation for your server to determine whether it supports cloning.

To Clone Server Settings to Another Server

1. In the Sun ONE Server Console navigation tree, select a reference server, the server that has the settings you want to replicate on other servers of the same type.

2. From the Object menu, choose Clone Server.

3. In the Select Target Servers for Cloning window, select the servers that you want to copy the reference server's settings to.

4. Click OK.

Removing a Server Instance

You can remove an instance of any server, other than Administration Server, from the navigation tree. Removing a server instance is useful when you no longer need to manage a particular server instance, but want to continue creating or using servers of the same type. When you remove an instance, all configuration settings and user data for that instance are deleted.

To Remove a Server Instance

1. In the navigation tree, select the server instance you want to remove.

2. From the Object menu, choose Remove Server.

Uninstalling a Sun ONE Server

If you no longer want to create or use any instances of any particular server, you can uninstall the server. This is different from removing a server instance since all program files are deleted when uninstalled. For more information on uninstallation, refer to "Uninstallation," on page 23.

Merging Configuration Data From Two Directory Servers

You can use Sun ONE Server Console's Merge Configuration Directory utility to merge the contents of two configuration directories. During a merge operation, the contents of a server group in one configuration directory are copied into a new server group in another configuration directory. No files are transferred during a Merge Configuration Directory operation; the destination configuration directory is simply updated to include information from the source.

The Merge Configuration Directory utility is useful if you've installed and deployed a number of Sun ONE servers, and now find it necessary to merge new data into an existing configuration directory.

For example, you may wish to test out a new product before deployment. Rather than make major changes to an existing configuration directory, you can try the product with a pilot instance of Directory Server, using just the new data required to configure the pilot.

This way, you can make adjustments to the new instance's configuration without having an impact on other server instances or the existing directory. Once you're satisfied with the settings in the pilot configuration directory, you can merge its configuration data into the configuration directory that's already deployed.

When merging configuration information, you copy from a source to a destination. In the example just described, the source is the pilot Directory Server instance with the new configuration data, and the destination is the existing Directory Server instance with current configuration data.

Figure 4-2 shows what two configuration directories might contain before they are merged.

Figure 4-2 Before Merging Two Configuration Directories

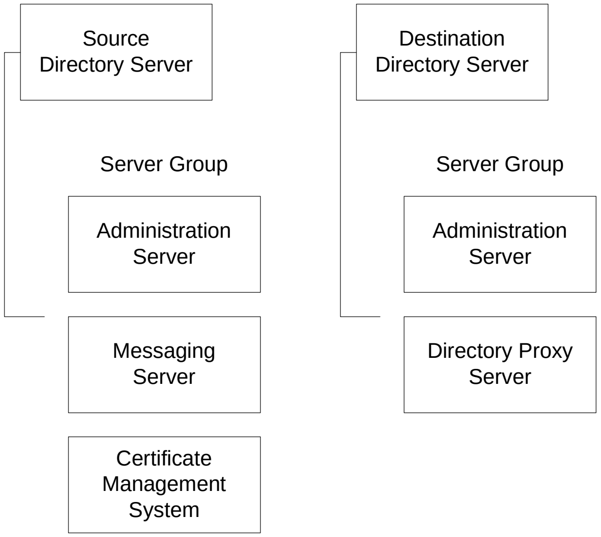
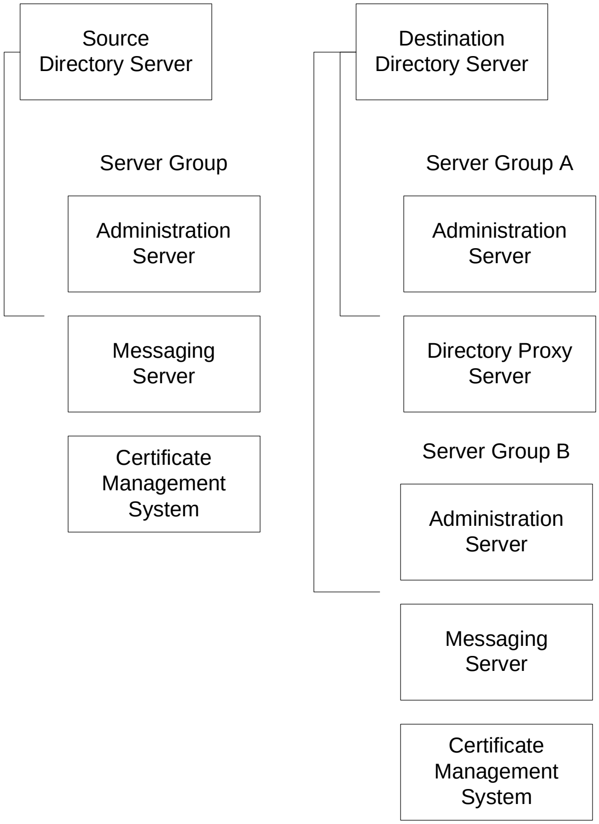


Figure 4-3 shows what the same two configuration directories would contain after you merged them.

Figure 4-3 After Merging Two Configuration Directories



When you have finished using the Merge Configuration Directory utility, you can safely remove your source configuration directory.

CAUTION Do not remove your source configuration directory until you have merged all data to the destination. Once you remove the source directory, you cannot restore it.

To Merge Configuration Data From Two Directory Servers

1. In the navigation tree, select the server group containing the source configuration directory.

2. From the Object menu, choose Merge Configuration.

3. In the Merge Configuration Directory Server Information window, enter information about the configuration directory into which you want to merge the source data:

   Destination Domain. Enter the domain name for the configuration directory that you want to merge into. Example: `example.com`

   Destination LDAP Host. Enter the host name for the configuration directory you specified above. Example: `eastcoast.example.com`

   Destination LDAP Port. Enter the port number for the existing configuration directory. Example: `389`

   Secure Connection. Check this box if the configuration directory uses the Secure Sockets Layer (SSL) protocol on the port specified above. Make sure that SSL is enabled on the destination configuration directory before selecting this option.

   Destination LDAP Bind DN. Enter the distinguished name for a user who has access to the destination configuration directory. Example: `cn=Barbara Jones, ou=Administration, o=Example Corporation, c=US`.

   Destination LDAP Bind Password. Enter the password for the user specified by the Destination LDAP Bind DN.

After you merge the configuration directories, the affected server instances use the destination directory you specify. If you want the instances to switch back to the original configuration directory, you must manually modify the local configuration files. See “Changing the Host or Port Number,” on page 106 for more information.

Chapter 5

User and Group Administration

Sun ONE Server Console allows you to create, locate, and manage user and group information from any system in your enterprise.

This chapter contains the following sections:

- Interacting with Directory Server
- Creating New Directory Entries
- Modifying Existing Directory Entries

Chapter 9, “Access Control” shows you how to work with user and group information when setting access privileges and other security information.

Interacting with Directory Server

When you use Sun ONE Server Console to create or modify users and groups, you make changes in the user directory, a subtree (suffix) of Directory Server. These changes affect all applications that use Directory Server. For information on how Sun ONE Server Console uses the data stored in the user directory, see Chapter 1, “Sun ONE Server Console and Administration Server.”

Using Distinguished Names

A distinguished name (DN) is a text string that identifies a specific directory branch or entry. Each user and group in your enterprise is represented in the Directory Server by a DN. Whenever you make changes to user and group information in the Directory, you use distinguished names (DNs). For example, you need to specify a DN each time you perform one of the following operations:

- Create or modify directory entries

- Set up access controls
- Set up user accounts for applications such as mail or publishing

From the Sun ONE Server Console "Users and Groups" tab, you can create, select, and use directory entries.

Distinguished Names, Attributes, and Syntax

This section presents a brief summary of distinguished names, attributes, and syntax information.

Distinguished Names

A *distinguished name* (DN) is the string representation of an entry's name and location in an LDAP directory. A DN describes a path to a directory entry. Each DN is made up of a number of components called *relative distinguished names* (RDNs). Each RDN identifies a specific entry in the directory. In order to ensure that every directory entry is unique, LDAP dictates that a single parent entry cannot have two identical RDNs below it.

Customarily, a DN for a user or group contains at least three types of RDN:

- A user name, user ID, or group name (identified by the `cn` or `uid` keyword)
- An organization name (identified by the `o` keyword)
- One or more domain name components (identified by the `dc` keyword). Example: `example.com` contains two domain name components: `example` and `com`.

Other common RDNs are organizational unit (`ou`), state (`st`), and country (`c`).

The exact composition of a DN depends on the structure of the directory. Most directories are organized by more categories than just country designations and organization names. As a result, the DNs used to identify entries are longer and contain more specific RDNs. For example, the DNs for three employees or users in the same company might look like this:

```
cn=Ben Hurst, ou=Operations, o=Example Corp, st=CA, c=US
```

```
cn=Jeff Lee, ou=Marketing, o=Example Corp, st=CA, c=US
```

```
cn=Mary Smith, ou=Sales, o=Example Corp, st=MN, c=US
```

In these examples, all three users work in different departments or organizational units (`ou`) and for the same company or organization (`o`), Example Corp. The third user works in a different state (`st`) from the first two users.

LDAP allows organizations and organizational units to contain other organizations and organizational units, allowing for the representation of complex enterprises. For example, the DN for a group within a large corporation might look like this:

```
cn=Technical Publications, ou=Super Server Group, ou=Server
Division, o=Example Corporation, o=MegaCorp, dc=megacorp, dc=com
```

Table 5-1 contains a list of common RDN keywords.

Table 5-1 Common RDN Keywords Used in DNs

RDN Keyword	Meaning in a DN	Description

`c`	country	Country in which the user or group resides. Examples:
		`c=US`
		`c=GB`

`cn`	common name or full name	Full name of person or object defined by the entry. Examples:
		`cn=Wally Henderson`
		`cn=Database Administrators`
		`cn=printer 3b`

`dc`	domain component	Part of a DNS domain. This keyword is typically used at the top levels of a directory tree. For example, a user in the `example.com` domain might have the following DN: `cn=Barbara Jones,ou=Engineering, dc=example, dc=com`

`l`	locality	Locality in which the user or group resides. This can be the name of a city, country, township, or other geographic regions. Examples:
		`l=Tucson`
		`l=Pacific Northwest`
		`l=Anoka County`

Table 5-1 Common RDN Keywords Used in DNs *(Continued)*

RDN Keyword	Meaning in a DN	Description

`o`	organization	Organization to which the user or group belongs. Examples:
		`o=Sun ONE Software`
		`o=Public Power & Gas`

`ou`	organizational unit	Unit within an organization. Examples:
		`ou=Sales`
		`ou=Manufacturing`

`sn`	surname	User's last name. Example: `sn=Henderson`

`st`	state or province	State or province in which the user or group resides. Examples:
		`st=Iowa`
		`st=British Columbia`

Keep in mind that the DNs you specify when using Sun ONE Server Console must reflect the types of data in your user directory. For information on setting up the user data in your Sun ONE Directory Server refer to the product documentation under `http://docs.sun.com/db/prod/s1dirsrv`.

Attributes

Directory attributes hold descriptive information about an entry. For example, a user entry might have attributes for a user ID, email address, given name, and password.

Table 5-2 contains a list of common user and group directory attributes.

Table 5-2 Common User and Group Directory Attributes

Attribute Keyword	Attribute Name	Description
`givenName`	given name	User's first name.
`mail`	email address	User's or group's email address.

Table 5-2 Common User and Group Directory Attributes *(Continued)*

Attribute Keyword	Attribute Name	Description
`streetAddress`	street	Street number and address of user or group defined by the entry. Example: `street=12 Main Street`
`telephoneNumber`	telephone	User's or group's telephone number. Example: `(800) 555-9SUN`

`title`	title	User's job title. Examples:
		`title=writer`
		`title=manager`

`uid`	user ID	Name that uniquely identifies the person or object defined by the entry.
`userPassword`	password	A user's password.

A user entry can include many more attributes than those listed above. In addition, you can create new attributes to meet your company's needs.

DN and Attribute Guidelines and Syntax

As you create, select, and use directory entries, follow these guidelines:

Separate RDNs with a comma. If an RDN value contains a comma, enclose the part of the name that uses the comma in double-quotation marks or escape it with a backslash. For example, to include the string `Ace Industry, Corp` in a DN, use the form:

```
o="Ace Industry, Corp", c=US
```

You may achieve the same effect using:

```
o=Ace Industry\, Corp, c=US
```

When schema checking is turned on, attributes must match directory schema. If you are using Sun ONE Directory Server and schema checking is turned on, use RDN keywords and attributes that can be recognized by the Directory Server and are allowed by the entry's object classes. If schema checking is turned off, you can use all attributes, regardless of an entry's object classes.

Specify RDNs in the same sequence or path. It is important to remember that a DN represents a path through a directory tree. If RDN keywords are not specified in the appropriate order, Directory Server may not be able to locate an entry. For example,

```
cn=Ralph Swenson, ou=Accounting, o=Example Corp, c=US
```

is not the same as

```
cn=Ralph Swenson, o=Example Corp, ou=Accounting, c=US
```

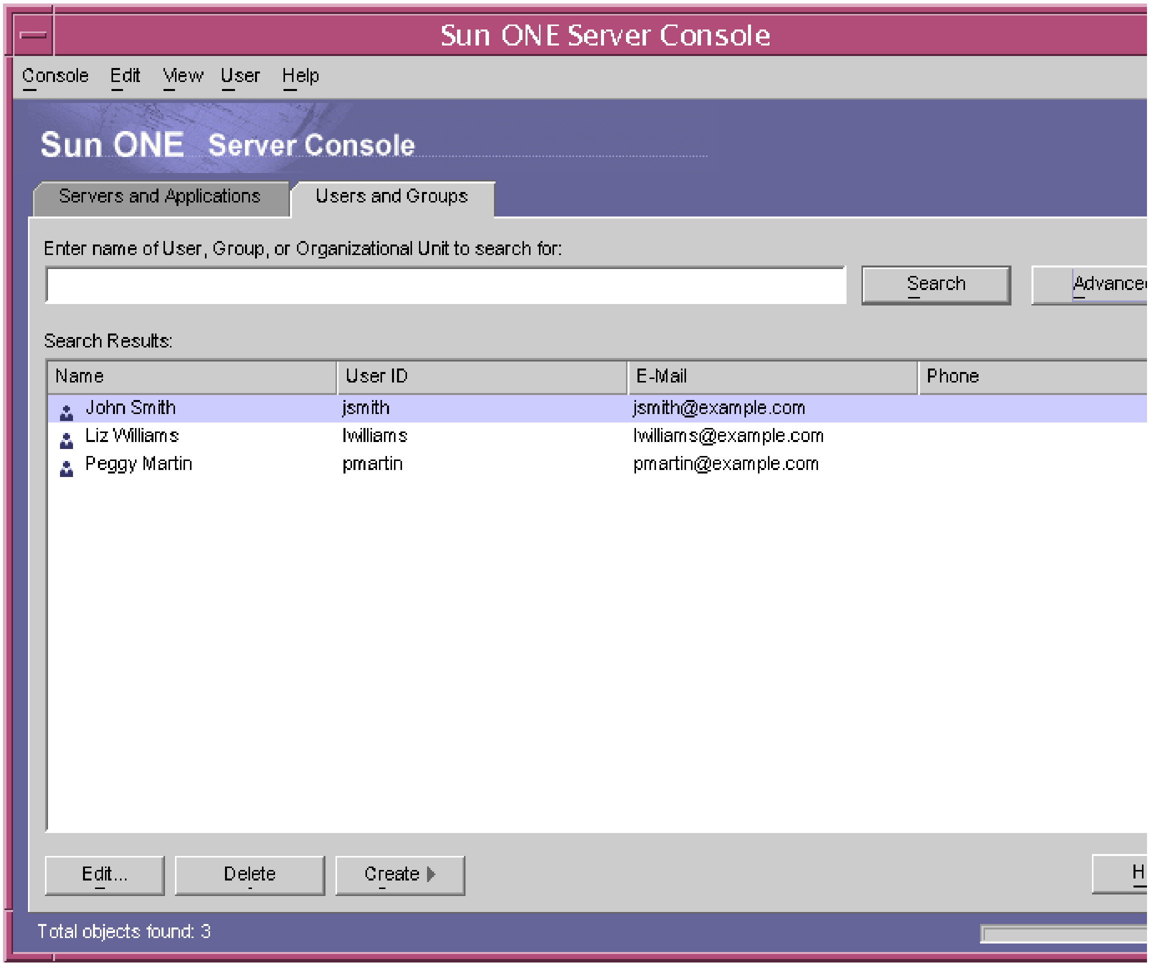
because the organizational unit (`ou`) and organization (`o`) keywords are not listed in the same order.

User IDs must be unique. If duplicate user IDs exist in your directory, users with those IDs cannot subsequently authenticate to the directory. Exercise caution when using the `ldapmodify` command line utility to create users, since the utility does not check for duplicate user IDs unless an attribute uniqueness plug-in is enabled in the directory for the user ID attribute.

Locating a User or Group in the Directory

You can use the “Users and Groups” Search function to locate directory entries. Initially, the function is set to search within the default user directory. If you do not want to use the default user directory, you can manually change to another one. See “Choosing a Different Directory to Search”, on page 65 for more information.

Figure 5-1 Sun ONE Server Console User and Groups Tab



Name	User ID	E-Mail	Phone
John Smith	jsmith	jsmith@example.com	
Liz Williams	lwilliams	lwilliams@example.com	
Peggy Martin	pmartin	pmartin@example.com	

Total objects found: 3

To Locate Users or Groups in the Directory

1. In Sun ONE Server Console, click the “Users and Groups” tab.

2. Specify your search criteria in one of these ways:

   To find specific entries, enter all or part of a user, group, or organizational unit name in the text entry box. For example, entering `John Swanson` returns any entries with DNs containing “John Swanson” while entering `John` returns all entries with DNs contains the word “John.”

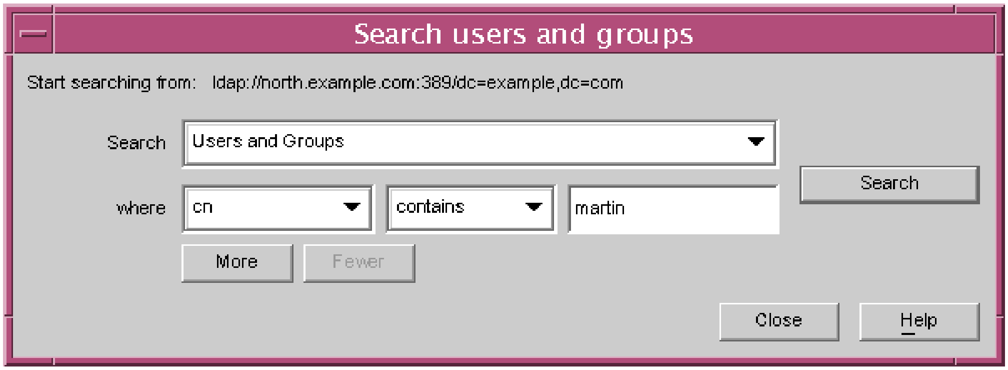
   To see all the entries currently stored in your directory, leave the Search field blank or enter an asterisk (*). Keep in mind that retrieving all entries in a large database can take a long time.

   To specify more focused search criteria, click the Advanced button. In the “Search users and groups” dialog box, enter the following information:

   Search. Specify where to perform the search by choosing Users, Groups, Users and Groups, or Administrators. The part of the subtree to search is specified at the top of the dialog box.

   Where. First choose an attribute, and then choose a search operator and type in a term.

Figure 5-2 Searching for User and Groups



Search users and groups

Start searching from: ldap://north.example.com:389/dc=example,dc=com

Search: Users and Groups

where: cn contains martin

3. Click Search.

   The search results are displayed in the list box.

NOTE For performance reasons, the Console for Directory Server displays only 5000 results, even when you have configured the Directory Server to return more than 5000 results.

Choosing a Different Directory to Search

When you use the Advanced Users and Groups Search function, the URL for the default user directory appears above the text entry box (see Figure 5-2). Initially, all searches are performed in this user directory. If you need to search a different user directory, you can choose one other than the default.

To Change the Directory to Search

1. In Sun ONE Server Console, click the “Users and Groups” tab.

2. From the User menu, choose Change Directory.

3. In the Change Directory dialog box, provide user directory information:

   User Directory Host. Enter the fully qualified host name where the user directory is installed.

   User Directory Port. Enter the port number used to connect to the user directory.

   Secure Connection. Check this box if the port number entered above is for use with the Secure Sockets Layer (SSL) protocol. Make sure that the port is configured to support SSL before selecting this option.

   User Directory Subtree. Enter the DN of the user directory subtree to search in. For example, to search all user entries in your organization, you might enter `dc=example,dc=com`. To search within the sales force, you might enter `ou=sales, dc=example,dc=com`.

   Bind DN. Enter the distinguished name of a user authorized to search entries in the user directory.

   Bind Password. Enter the password for the user specified by the Bind DN.

4. Click OK.

Creating New Directory Entries

From the Sun ONE Server Console "Users and Groups" tab, you can add or modify a user, group, or organizational unit. Alternatively, you can perform these directory operations from the command line using tools such as `ldapmodify`(1) on Solaris systems.

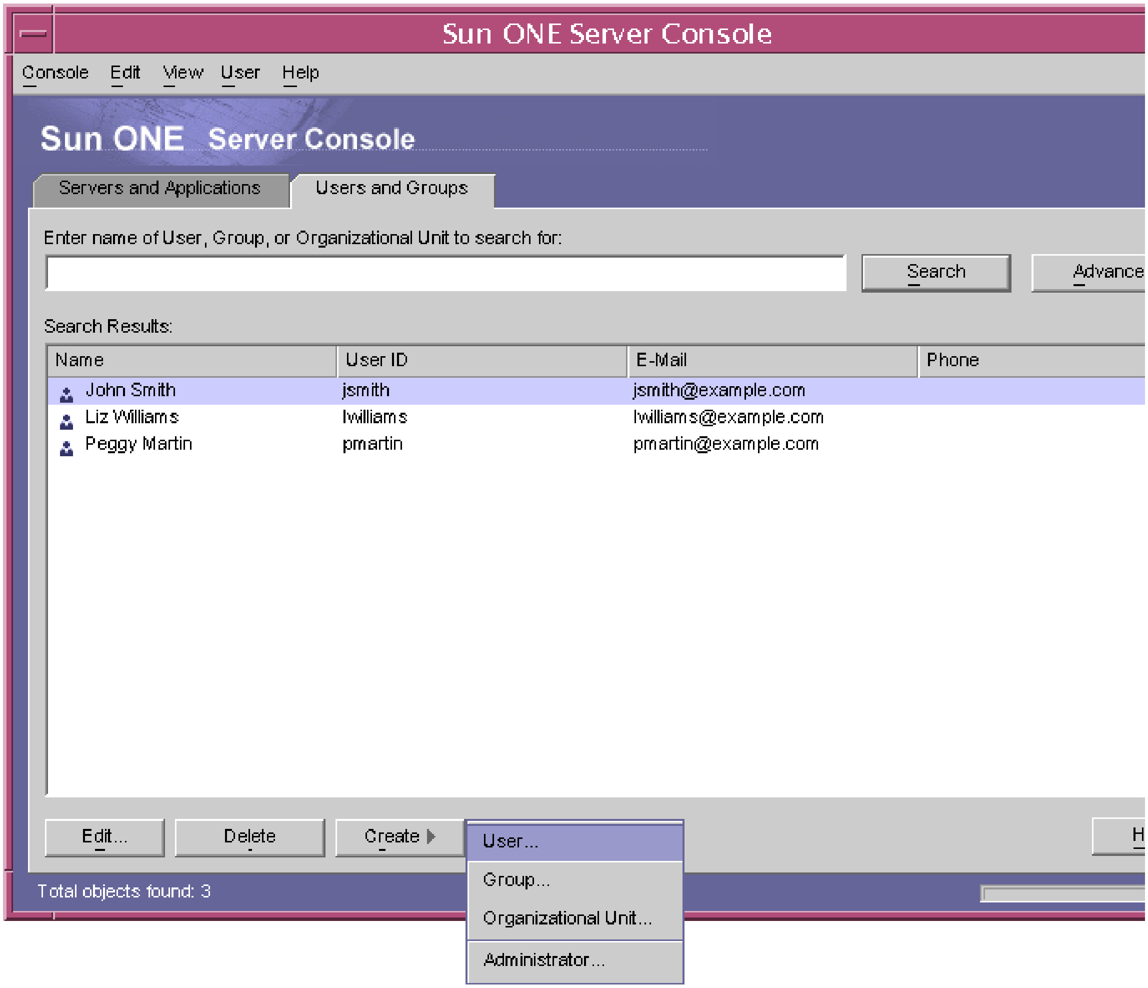
Users

A user entry contains information about an individual person or resource in the directory. For example, you can create user entries for `John Smith`, `Printer 3B`, or `Conference Room 25`.

To Create a New User Entry in the Directory

1. In Sun ONE Server Console, click the "Users and Groups" tab.
2. Click the Create button and then choose User. Alternatively you can open the User menu and choose Create > User.

Figure 5-3 Creating a User



Enter name of User, Group, or Organizational Unit to search for:

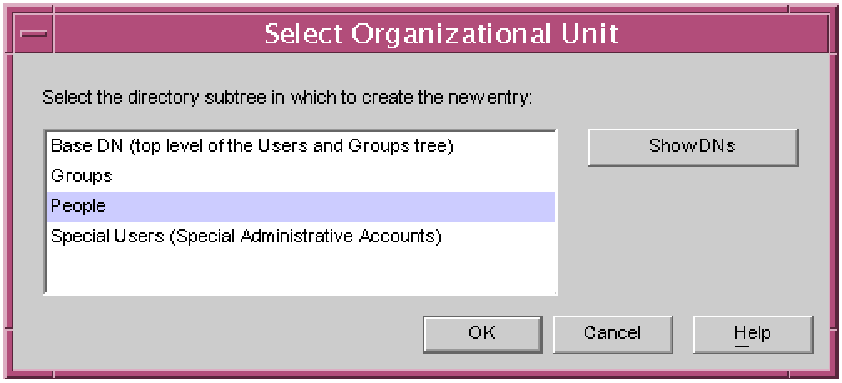
Search Results:

Name	User ID	E-Mail	Phone
John Smith	jsmith	jsmith@example.com	
Liz Williams	lwilliams	lwilliams@example.com	
Peggy Martin	pmartin	pmartin@example.com	

Total objects found: 3

3. In the Select Organizational Unit dialog box, select the organizational unit (ou) or top entry of the subtree to which the user belongs, and then click OK.

Figure 5-4 Selecting the Organizational Unit



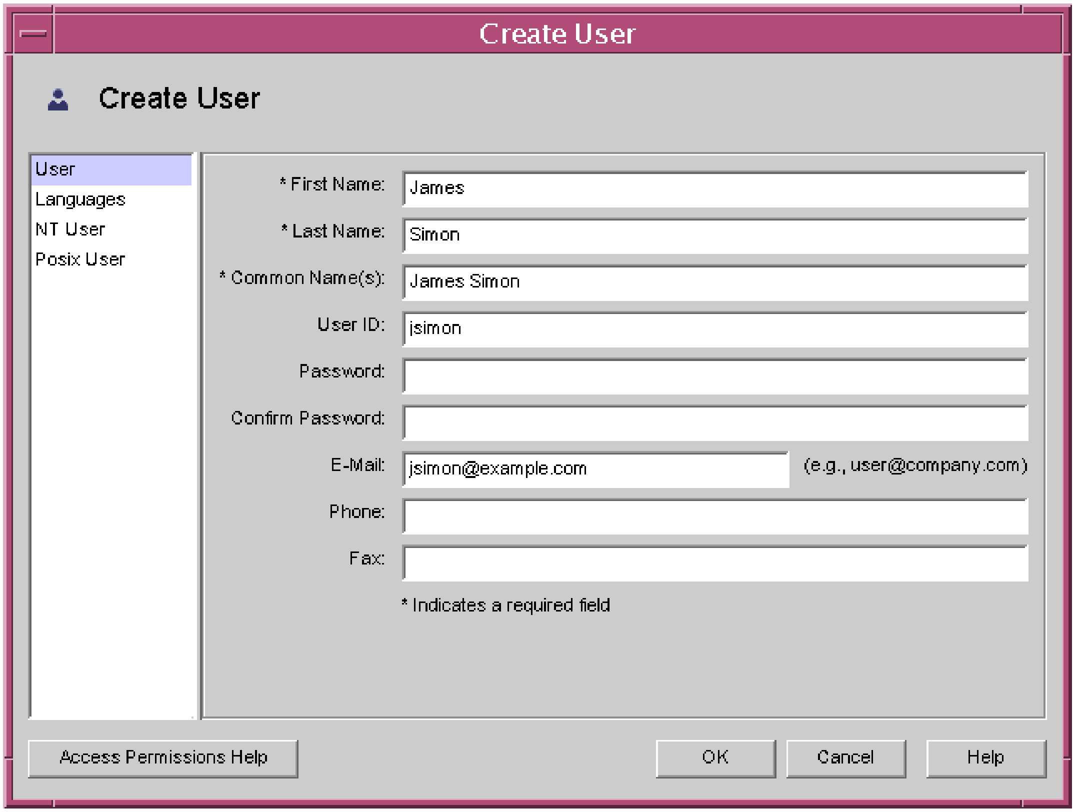
Select Organizational Unit

Select the directory subtree in which to create the new entry:

- Base DN (top level of the Users and Groups tree)
- Groups
- People
- Special Users (Special Administrative Accounts)

4. In the Create User window, enter user information:

Figure 5-5 Entering User Information



* First Name: James

* Last Name: Simon

* Common Name(s): James Simon

User ID: jsimon

Password:

Confirm Password:

E-Mail: jsimon@example.com (e.g., user@company.com)

Phone:

Fax:

* Indicates a required field

First Name. Enter the user's first name.

Last Name. Enter the user's last name (surname).

Common Name. This is the user's full name. It is automatically generated based on the First Name and Last Name entered above. You can edit this name as necessary.

User ID. When you enter a first and last name, the user ID is automatically generated. You can replace this user ID with one of your choosing. The user ID must be unique from all other user IDs in the directory.

Password. (Optional) Enter the user's password. Alphanumeric characters, spaces, and punctuation marks are all acceptable.

Confirm Password. If you entered the user's password, enter it again to confirm.

E-Mail. (Optional) Enter the user's mail address. If the user has multiple mail addresses that you want to store in the same attribute, separate them with commas. For example: `jdoe@example.com, jane.doe@example.net`

Phone. (Optional) Enter the user's telephone number. If the user has multiple telephone numbers that you want to store in the same attribute, separate them with commas. For example: `(800)555-9SUN, (650)960-1300`

Fax. (Optional) Enter the user's fax number. If the user has multiple fax numbers that you want to store in the same attribute, separate them with commas.

5. If you want to specify language-related information, click the Languages tab. From the drop-down list in the Languages panel, select the user's preferred language, and then enter language-related information:

   First Name. Enter the user's first name in the selected language.

   Last Name. Enter the user's last name (surname) in the selected language.

   Common Name. This is the user's full name in the selected language. It is automatically generated based on the First Name and Last Name entered above. You can edit this name as necessary.

   Phone. Enter the user's telephone number. If the user has multiple telephone numbers that you want to store in the same attribute, separate them with commas. For example: `(800)555-9SUN, (650)960-1300`

   Pronunciation. If the selected language is commonly represented phonetically, additional fields are displayed. Enter the phonetic representation for the user's first, last, and common name.

6. If you want to specify UNIX or Windows specific attributes, click the NT User or Posix User tab. For more information, see "Specifying UNIX and Windows Systems Options." on page 72.

7. Click OK.

The User's Preferred Language

Sometimes a user's name can be more accurately represented using a character set other than that of the default language. For example, Noriko's name is Japanese, and she has indicated on her hiring forms that she prefers when Japanese characters represent her name. You can select Japanese as her preferred language so that her name is displayed in Japanese characters, even when a user's default language is English.

To indicate a user's preferred language, follow the instructions in Step 5 of the procedure "To Create a New User Entry in the Directory," on page 66.

Administrators

During installation, you are asked to enter a user name and password for the *Configuration Administrator*, the user authorized to access and modify the entire configuration directory. The Configuration Administrator entry is stored in the directory under the following DN:

```
uid=userID, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot
```

During installation, the Configuration Administrator's user name and password are used to automatically create the *Administration Server Administrator*. This user can perform a limited number of tasks, such as starting, stopping, and restarting servers in a local server group. The Administration Server Administrator is created for the purpose of logging into Sun ONE Server Console when the Directory Server is not running.

The Administration Server Administrator does not have an LDAP entry; it exists only as an entity in a local configuration file stored at:

```
ServerRoot/admin-serv/config/admpw
```

Even though they are created at the same time during installation, and are identical at that time, the Configuration Administrator and Administration Server Administrator are two separate entities. If you change the user name or password for one, Sun ONE Server Console does not automatically make the same changes for the other.

For more information on modifying the Configuration and Administration Server Administrators, see "Modifying Existing Directory Entries," on page 84.

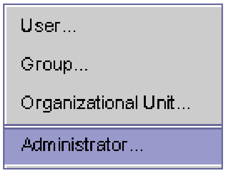
To Create an Administrator

The administrator user you create has the same rights as the Configuration Administrator created during installation, and the administrator user entry is located in the same subtree as that of the Configuration Administrator.

1. In Sun ONE Server Console, click the "Users and Groups" tab.

2. Click the Create button and then choose Administrator.

Figure 5-6 Creating an Administrator User



Alternatively, you can open the User menu and choose Create > Administrator.

3. In the Create Administrator window, enter the appropriate user information.

   The requested information is exactly the same as in the Create User dialog box, except that Password is a required field. For more information, refer to "To Create a New User Entry in the Directory," on page 66.

Specifying UNIX and Windows Systems Options

You can enable additional user configuration panels to store UNIX and Windows user information in the directory. You can use these panels to specify the options and attributes to synchronize with your operating system. There are two panels you can enable: NT User and Posix User.

By default, you must enable these panels for each individual user. If you want to enable these panels automatically for every new user, you can do so by modifying the configuration directory. Once you have enabled these panels, you can use them to set Windows and UNIX systems options and attributes.

The following procedures show you how to enable these panels and modify Windows and UNIX systems options and attributes.

To Enable UNIX and Windows Systems Panels for an Individual User

1. In the Create User window, click the NT User or Posix User tab.

   The appropriate panel appears.

2. Enable the fields in the panel.

   To enable the NT User fields, select "Enable Windows NT user attributes."

   To enable the Posix User fields, select "Enable Posix user attributes."

To Enable UNIX and Windows Systems Panels for All New Users

1. Open your Directory Server management window.
2. Click the Directory tab and click `NetscapeRoot` in the navigation tree.
3. Click to open your administration domain, and then expand GlobalPreferences > *AdminDomain* > Admin > 4.0.
4. Click the defaultObjectClassesContainer folder, and then double-click "user" in the right-hand panel.
5. Select "nsdefaultobjectclass," then, from the Edit menu, choose Add Value.

   A blank field appears. If you are enabling both the Windows NT and Posix/UNIX panels, choose Add Value a second time to create another blank field.
6. Enter the appropriate object class name in the field.

   To enable the NT User panel, enter `ntUser`. To enable the Posix User panel, enter `posixUser`.
7. Click OK.

To Set UNIX and Windows Systems Options and Attributes for a New User

1. Follow steps 1-5 of "To Create a New User Entry in the Directory." beginning on page 66.
2. If you want to store Windows specific user information in the directory, click the NT User tab, enable the fields by selecting "Enable Windows NT user attributes," and then enter the following information:

   NT User ID. Enter the user's NT login name.

   Delete NT Account If Person Deleted. (Optional) Checking this box does not delete the user.

Comment. (Optional) Enter a descriptive comment about this user.

User Profile Path. (Optional) Enter the path to this user's profile. Use the Windows network path format. For example: `\\server\profiles\josu`.

Logon Script. (Optional) Enter the path to the user's logon script. This path is relative to the system's logon script path. For example, if the system path is `\\server\logon`, you might enter `writers.bat` or `writers\josu.cmd` depending on where you store your user scripts.

Home Drive. (Optional) Use the drop-down list to choose the drive on which this user's home directory is located.

Home Directory. (Optional) Enter the path to this user's home directory. Use the Windows network path format or an absolute path. For example, you can enter either `\\server\users\josu` or `C:\user profiles\josu`.

Logon Server. (Optional) Enter the path to the server on which this user's logon script is stored. Use the Windows network path format.

Logon Hours. (Optional) Click to set the hours during which this user can log on.

User Workstations List. (Optional) Enter the computers from which this user can log on.

Change. (Optional) Click to change the date and time at which the user's account expires.

3. If you want to store UNIX specific user information in the directory, click the Posix User tab, enable the fields by selecting "Enable Posix user attributes," and then enter the following information:

   UID Number. Enter the user's UNIX ID number.

   GID Number. Enter the user's UNIX group ID number.

   Home Directory. Enter the path to the user's home directory. For example, `/home/josu`.

   Login Shell. (Optional) Enter the path to the user's login shell. For example, `/bin/bash`.

   Gecos. (Optional) The value of this user's `pw_gecos` entry in `/etc/passwd`.

4. Click OK.

Groups

A group consists of users who share a common attribute or are part of a list. For example, you might set up a group called Sales consisting of all users whose entries contain the attribute `ou=Sales`. Sun ONE Directory Server supports three types of groups: static, dynamic, and certificate. Each group differs in the way in which users, or *members,* are added to it. The following descriptions explain this.

A *static group* consists only of users that have been added to it. It is called static because it doesn't change unless you add a user to it or delete a user from it. For example, if you create a static group called Marketing, none of the users who have the attribute `department=marketing` in their entry are members of the Marketing group until you explicitly add each one to the group.

TIP For high performance, avoid huge static groups. Use roles instead.

One special static group is called the *Configuration Administrators group*. It is automatically created and populated when the configuration directory is installed. Members of the Configuration Administrators group have unrestricted access to the configuration directory. The group is stored in the configuration directory under the following DN:

```
ou=Groups, ou=TopologyManagement, o=NetscapeRoot
```

Initially, the Configuration Administrator is the only member of the Configuration Administrators group. If he wants to give additional users his level of administrative privilege, he can do so by adding them as members of the group. These users can access the configuration directory in the same way as the Configuration Administrator. Any member of the Configuration Administrators group can add additional members.

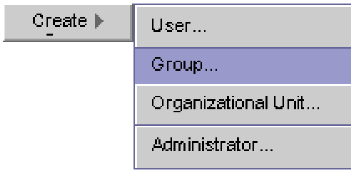
A *dynamic group* automatically includes users based on one or more attributes in their entry. For example, you can create a dynamic group called California Sales that automatically includes any entry containing the attributes `st=California` and `department=sales`. These attributes are specified as part of an LDAP URL. Whenever you search for members of the California Sales group, the results contain all entries located by the URL.

A *certificate group* includes all users who have a certificate containing a common attribute. For example, you can create a certificate group called California Western Sales whose members share these attributes: `ou=Sales, ou=West, st=CA`. When an individual user logs on to a server, if all of these attributes are found in his certificate, the user is automatically recognized as belonging to the group. If the user's certificate does not contain these attributes, he is not recognized as a member of the California Western Sales group and does not receive the same access, privileges, or permissions as group members.

To Create a Static Group in the Directory

1. In Sun ONE Server Console, click the "Users and Groups" tab.
2. Click the Create button and then choose Group. Alternatively you can open the User menu and choose Create > Group.

Figure 5-7 Creating a Static Group

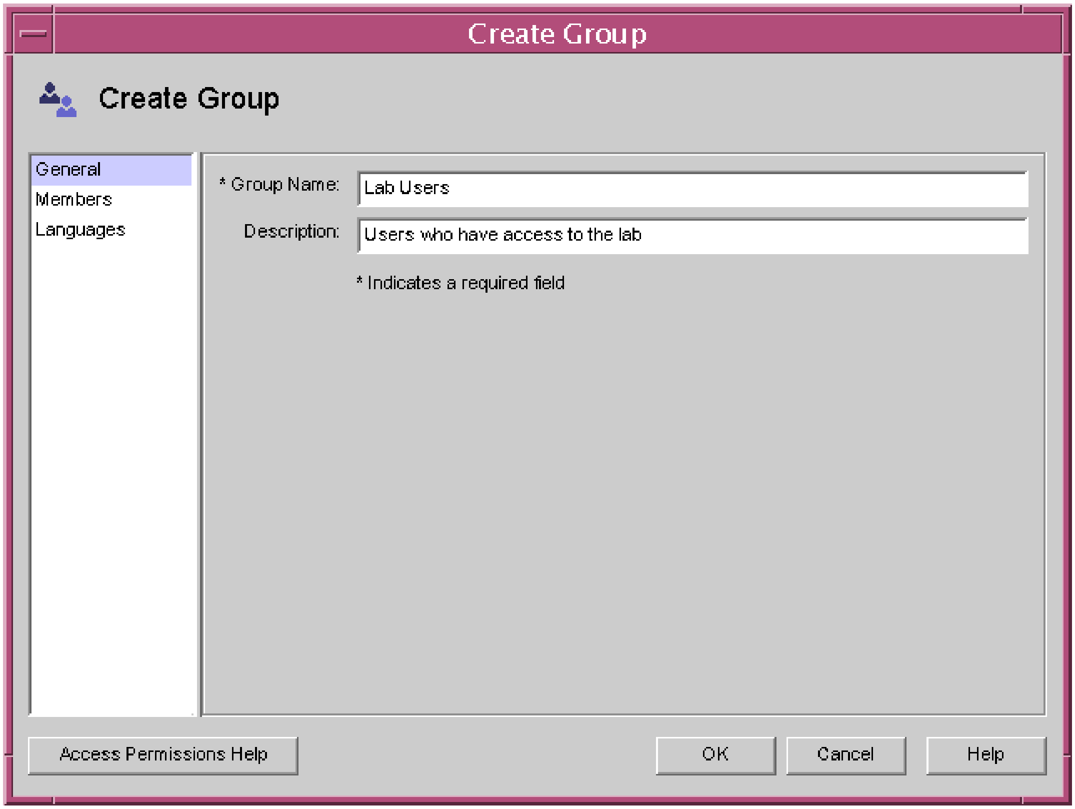


3. In the Select Organizational Unit dialog box, select the organizational unit(ou) to which the group belongs, and then click OK.
4. In the Create Group dialog box, enter group information:

   Group Name. Enter a name for the group.

   Description. (Optional) Enter a description to help you identify this group.

Figure 5-8 Entering Group Information



* Group Name: Lab Users

Description: Users who have access to the lab

* Indicates a required field

5. Create the group, or specify members for the group before creating it.

   If you want to create only the group now, and add group members later, click OK and skip the rest of this procedure.

   If you want to immediately add members to the group, click Members and then continue to the next step.

6. In the Members panel, click Add, and then use the Search dialog box to locate a user you want to add to the Members User ID list. Repeat this step until all the users you want to add to the group are displayed in the Member User ID list.

To Add Users to the Configuration Administrators Group

1. In Sun ONE Server Console, click the “Users and Groups” tab, and then choose Change Directory from the User menu.

2. In the Change Directory window, indicate the location of the user directory that contains the Configuration Administrators group:

   User Directory Host. Enter the fully qualified host name where the user directory is installed.

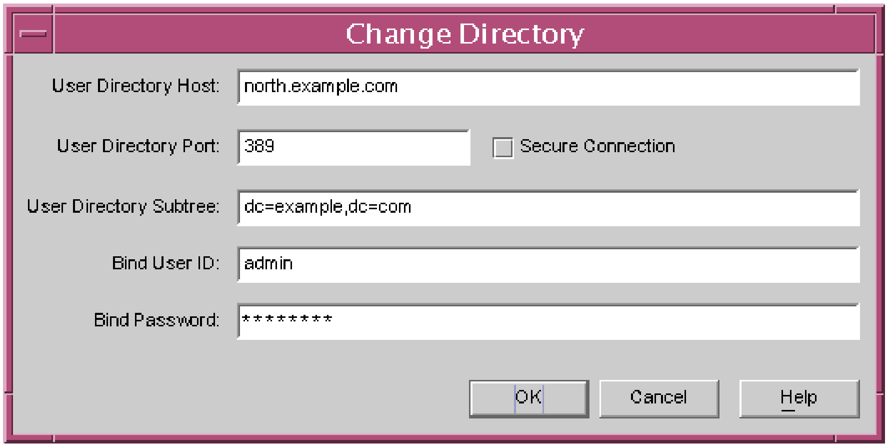
   User Directory Port. Enter the port number you want to use to connect to the user directory.

   User Directory Subtree. Enter `o=NetscapeRoot` to indicate where to find the Configuration Administrators group.

   Bind DN. Enter the DN of a user authorized to change entries in the user directory.

   Bind Password. Enter the password of the user directory administrator.

Figure 5-9 Change to the Directory Holding the Administrator Subtree



Change Directory

User Directory Host: north.example.com

User Directory Port: 389 □ Secure Connection

User Directory Subtree: dc=example,dc=com

Bind User ID: admin

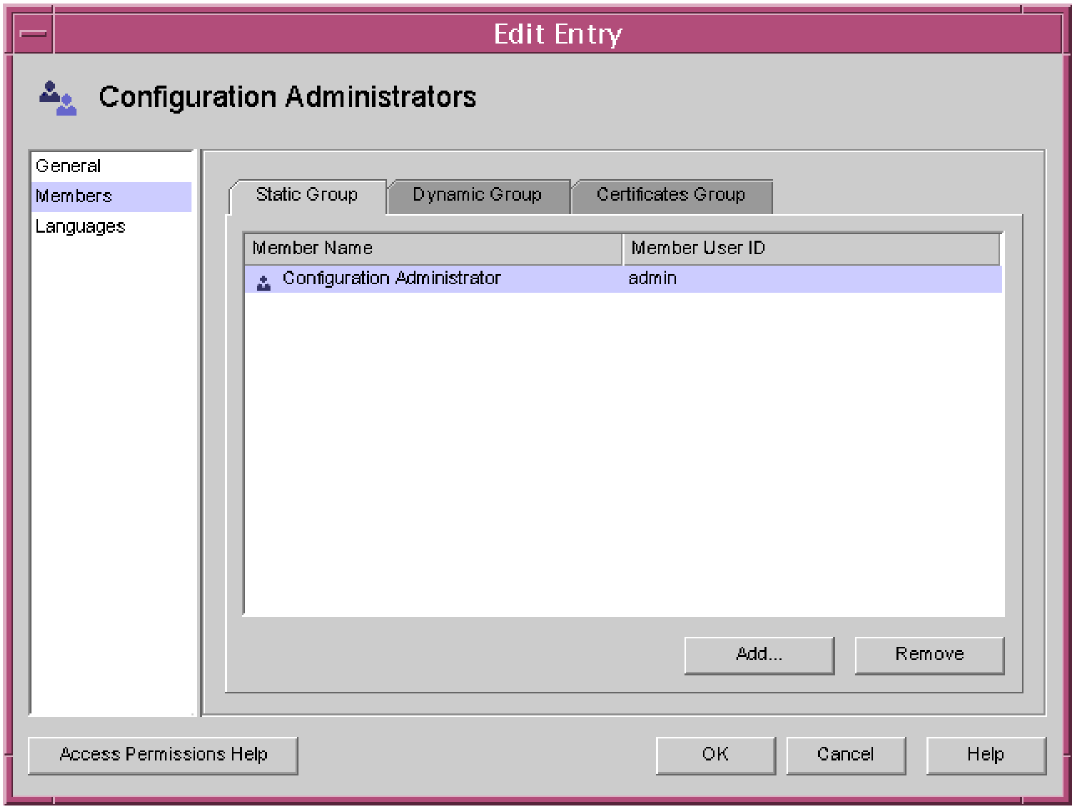
Bind Password: ********

3. Click OK.

4. Use the Search function to locate and highlight the Configuration Administrators group, and then click Edit.

5. In the Edit Group window, click Members.

Figure 5-10 Adding the User to the Administrator Group



Member Name	Member User ID
Configuration Administrator	admin

6. Click Add.

7. In the Search Users and Groups window, locate and select the user you want to add, and then click OK.

   Repeat this step until all the users you want to add to the group are displayed in the Members list, and then click OK.

To Create a Dynamic Group

1. In Sun ONE Server Console, click the “Users and Groups” tab.

2. Click the Create button and then choose Group. Alternatively you can open the User menu and choose Create > Group.

3. In the Select Organizational Unit dialog box, select the organizational unit (ou) to which the group is to belong, and then click OK.

4. In the Create Group dialog box, enter general group information.

   Group Name. Enter a name for the group.

   Description. (Optional) Enter a description to help you identify this group.

5. Click Members.

6. Click Dynamic Group, and then click Add.

7. Use the “Construct and Test LDAP URL” dialog box to specify the criteria for including users in the dynamic group.

   If you know the exact LDAP URL you want to use to include users in the group, enter it and skip to Step 10.

   The LDAP URL takes the form:

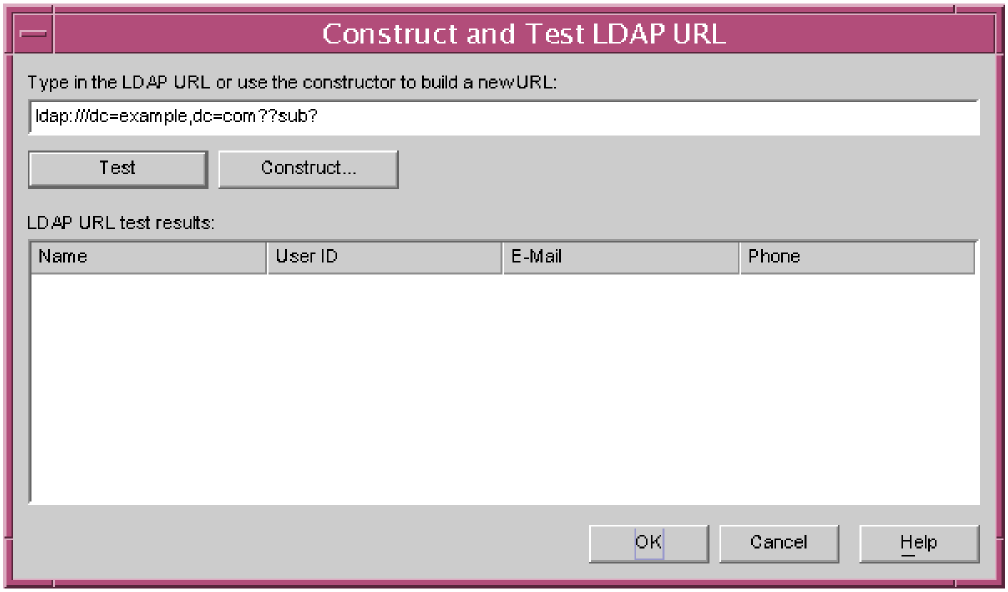
   `ldap:///o=`*base_suffix*`??sub?(`*RDN_or_attribute*`=`*value*`)`

   For example:

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

   If you want to interactively build an LDAP URL for including users in the group, click Construct.

Figure 5-11 Constructing the LDAP URL



Construct and Test LDAP URL
Type in the LDAP URL or use the constructor to build a new URL:
ldap:///dc=example,dc=com??sub?
Test
Construct...
LDAP URL test results:
Name
User ID
E-Mail
Phone
OK
Cancel
Help

8. In the Construct LDAP URL dialog box, provide search criteria:

   LDAP Server Host. Displays the fully qualified host name of the Directory Server in which you are searching.

   Port. Displays the port number for the listed LDAP Server Host.

   Base DN. Enter the base DN for from which to begin the search. Example: `ou=Marketing, o=Example Corp, c=US`

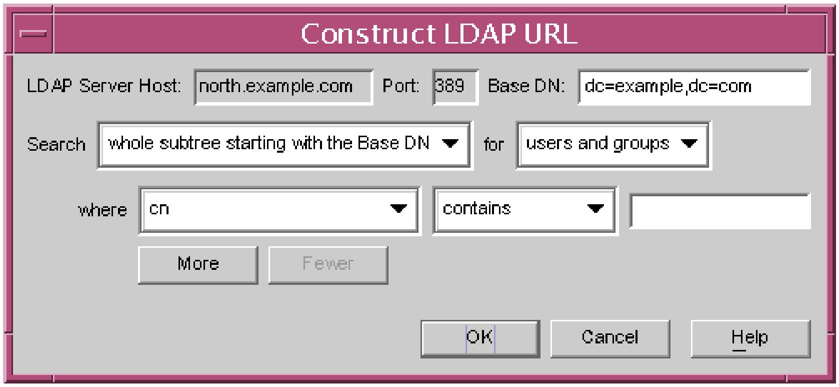
   Search. Specify the user directory subtree you want to search.

   for. Specify whether you want to search users, groups, or both.

   where. In the drop-down lists, first select an attribute, and then a search operator. In the last input field, enter a search string, and then click Search.

   More. If you want to specify more attributes to search for, click this button.

Figure 5-12 The Construct LDAP URL Dialog



LDAP Server Host: north.example.com Port: 389 Base DN: dc=example,dc=com

Search: whole subtree starting with the Base DN for: users and groups

where: cn contains

9. Click OK.

10. If you want to see a list of users and groups included in the dynamic group, click Test in the Construct and Test LDAP URL dialog box.

11. Click OK to confirm your acceptance of the LDAP URL and add it to the list used to include members in this dynamic group.

    If you want to create additional LDAP URLs for including members in this group, repeat steps 6 through 11.

To Create a Certificate Group

1. In Sun ONE Server Console, click the “Users and Groups” tab.

2. Click the Create button and then choose Group. Alternatively, you can open the User menu and choose Create > Group.

3. In the Select Organizational Unit dialog box, select the organizational unit (ou) to which the group belongs, and then click OK.

4. In the Create Group dialog box, enter group information:

   Group Name. Enter a name for the group.

   Description. (Optional) Enter a description that helps you identify this group.

5. Click Members

6. Click Certificate Group, and then click Add.

7. In the Certificate Group dialog box, fill in one or more of the following fields:

   Common Name. Enter the full name of the group. Example: `Database Administrators`.

   Organization. Enter the name of the organization the group belongs to. Example: `Operations Group`.

   Mail. Enter the street address for the group.

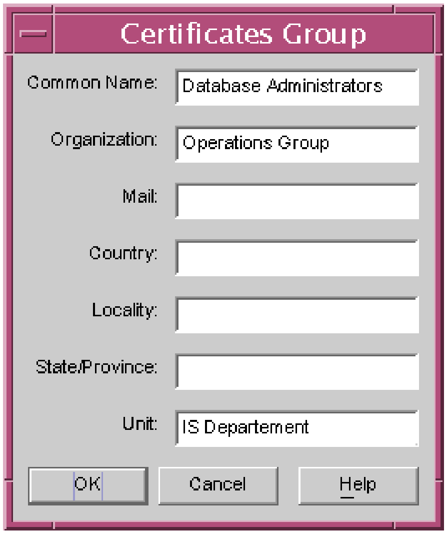
   Country. Enter the country code for the group.

   Locality. Enter the city name for the group's business.

   State/Province. Enter the state or province name for the group.

   Unit. Enter the name of the organizational unit that the group belongs to. Example: `IS Department`.

Figure 5-13 The Certificates Group Dialog



Common Name: Database Administrators

Organization: Operations Group

Mail:

Country:

Locality:

State/Province:

Unit: IS Departement

8. Click OK.

Organizational Units

An organizational unit can include a number of groups and usually represents a division, department, or other discrete business group.

When you create a new organizational unit, you add a branch to the directory. This is reflected through the use of an ou RDN. For example, if you create a new organizational unit called Accounting within the organizational unit West Coast, and your Base DN is `o=Example, c=US`, then the new organizational unit's DN is:

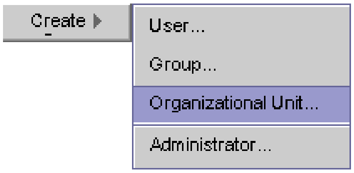
```
ou=Accounting, ou=West Coast, o=Example, c=US
```

To Create a New Organizational Unit

1. In Sun ONE Server Console, click the "Users and Groups" tab.

2. Click the Create button and then choose Organizational Unit. Alternatively, you can open the User menu and choose Create > Organizational Unit.

Figure 5-14 Creating an Organizational Unit



3. In the Select Organizational Unit dialog box, select the directory subtree in which to store the new organizational unit.

4. In the Create Organizational Unit dialog box, enter organizational unit information:

   Name. Enter a name for the organizational unit.

   Description. (Optional) Enter a description that helps you identify the organizational unit.

   Phone. (Optional) Enter a phone number where one can reach a contact person (such as an administrative assistant) for the organizational unit.

   Fax. (Optional) Enter a fax number where one can reach a contact person (such as an administrative assistant) for the organizational unit.

   Alias. (Optional) Enter another name, such as a nickname or acronym, that you might use in place of the Name entered above.

5. Click OK.

Modifying Existing Directory Entries

From the Sun ONE Server Console “Users and Groups” tab, you can change existing directory entries. Therefore, you can easily update user and group information whenever you need to.

Updating User and Group Entries

Before you can modify user or group data, you must first locate a user or group entry in the directory. See “Locating a User or Group in the Directory” on page 65 for more information on using the “Users and Groups” Search function to find directory entries.

Once you have located an entry, you can modify it or remove it. If you are working with a user entry, alternatively, you can change its password.

To Edit a User or Group Entry in the Directory

1. In the “Users and Groups” tab of Sun ONE Server Console, use the Search function to locate the user or group.
2. Once the user or group name appears in the search results list, select it, and then click Edit.
3. Modify user or group information as necessary, and then click OK.

To Change a User Password

1. In the “Users and Groups” tab of Sun ONE Server Console, use the Search function to locate the user.
2. Once the user appears in the search results list, select it, and then click Edit.
3. Enter the new password information:

   Password. Enter the new password. Alphanumeric characters, spaces, and punctuation marks are all acceptable.

   Confirm Password. Enter the password again to confirm.
4. Click OK for the change to take effect.

To Change the Configuration Administrator’s User Name or Password

1. In the “Users and Groups” tab of Sun ONE Server Console, click Advanced.

2. In the "Search users and groups" dialog box, enter search information.

   If you have never changed the Configuration Administrator's user name, enter the following information:

   Search. Select `Administrators` from the drop-down list.

   where. Select `cn` and `contains` from the drop-down lists and enter `Configuration Administrator` in the field.

   If you have changed the Configuration Administrator's user name, enter the following information:

   Search. Select `Administrators` from the drop-down list.

   where. Select `cn` and `contains` from the drop-down lists and enter the user name of the Configuration Administrator in the field.

3. Click Search.

   The results appear in the "Users and Groups" tab.

4. Click Close.

5. Select the Configuration Administrator from the list of search results, and then click Edit.

6. Enter the administrator's new user name and password:

   First Name. Enter the administrator's first name.

   Last Name. Enter the administrator's last name (surname).

   Common Name. This is the administrator's full name. It is automatically generated based on the First Name and Last Name entered above. You can edit this name as necessary.

   User ID. When you enter a first and last name, the user ID is automatically generated. You can replace this user ID with one of your choosing.

   Password. (Optional) Enter the new administrator's password. Alphanumeric characters, spaces, and punctuation marks are all acceptable.

   Confirm Password. If you entered a password, enter it again to confirm it.

7. Click OK.

8. If you bind to the directory as the Configuration Administrator when searching for users, Update your user directory information by completing these steps:

a. Click the “Users and Groups” tab of Sun ONE Server Console, and choose Change Directory from the User menu.

b. In the Change Directory window, update the Bind DN or user ID, and the Bind Password with the new information for the Configuration Administrator, and then click OK.

To Change the Administration Server Administrator’s User Name or Password

1. In the Sun ONE Server Console navigation tree, select the Administration Server instance that you want to change the administrator user name or password for.

2. Click Open to open the management window for the instance of Administration Server.

3. Click the Configuration tab.

4. In the Configuration tab, click the Access tab.

5. In the Access tab, enter information for the following fields:

   Username. Enter the user name for the Administration Server Administrator.

   Password. Enter the password for the Administration Server Administrator.

   Confirm Password. Enter the password again to confirm it.

   If you make an error while entering this information, you can click Reset to restore the original values for the fields.

6. Click Save to save the new Administration Server Administrator user name or password.

7. Restart the instance of Administration Server.

To Remove a User, Group, or Organizational Unit From the Directory

1. In the “Users and Groups” tab of Sun ONE Server Console, use the Search function to locate and highlight the user, group, or organizational unit you want to delete.

   If you are removing an organizational unit, you must first remove all users and groups belonging to it.

2. Click Delete.

3. Click OK when prompted to confirm the deletion.

Part 3

Using Sun ONE Administration Server

Chapter 6

Administration Server Basics

Sun ONE Administration Server processes requests for servers installed in a server group under the same root directory, and then starts the programs required to fulfill them. For a brief overview of Sun ONE Server Console architecture, see Chapter 1, "Sun ONE Server Console and Administration Server."

This chapter tells you how to perform basic Administration Server operations. It contains the following sections:

- Restarting Administration Server
- Stopping Administration Server
- Logging Options

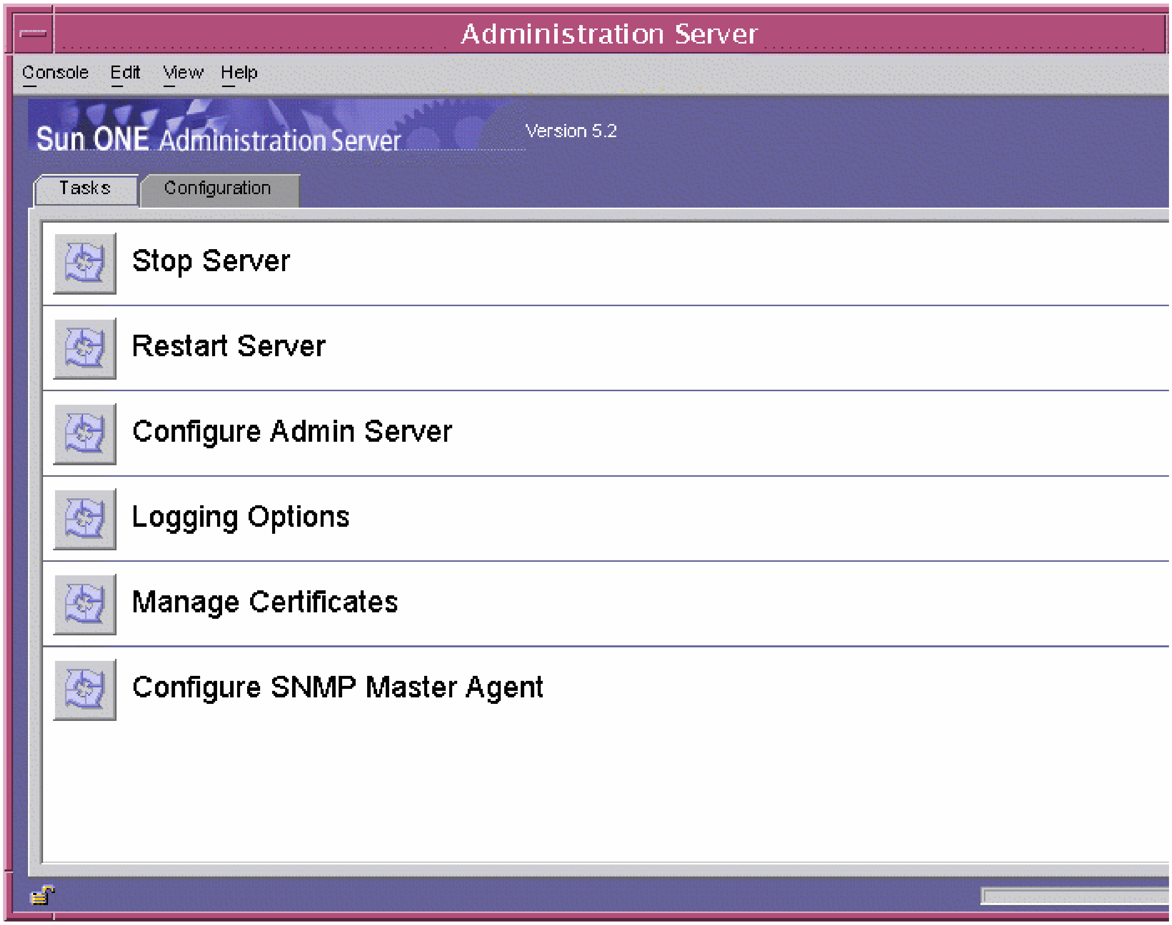
Restarting Administration Server

Sun ONE Administration Server automatically starts after installation. When you need to restart Administration Server, you can do so from Sun ONE Server Console or from the command line. On Windows systems, you can also restart the server from the Services control panel.

To Restart the Server From the Console

1. From the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to restart.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Tasks tab, and then choose Restart Server.

Figure 6-1 Administration Server Management Window



To Restart the Server From the Command Line

UNIX Systems

When using the Solaris packaged version, enter `/usr/sbin/mpsadmserver restart`.

When not using the Solaris packaged version, change to the server root directory, enter `./stop-admin`, and then enter `./start-admin`.

Windows Systems

1. Click Start, choose Run, and then enter the following to stop the server:

   ServerRoot/`stop-admin.cmd`

2. Click Start, choose Run, and then enter the following to start the server again:

   ServerRoot/`start-admin.cmd`

To Restart the Server From the Windows Control Panel

1. Open the Services control panel.

   Refer to Windows help for details on where to find the Services control panel.

2. Select Sun ONE Administration Server Version 5.2 from the list of services, click the Stop button, and then click the Start button.

3. Click Close to exit the Services control panel.

Stopping Administration Server

You can stop an instance of Administration Server from within Sun ONE Server Console or from the command line. On Windows NT, you can also stop the server from the Services control panel.

To Stop the Server From Sun ONE Server Console

1. From the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to stop.

2. Click Open to open the management window for the instance of Administration Server.

3. Click the Tasks tab, and then choose Stop Server.

To Stop the Server From the Command Line

UNIX Systems

When using the Solaris packaged version, enter `/usr/sbin/mpsadmserver stop`.

When not using the Solaris packaged version, change to the server root directory, and then enter `./stop-admin`.

Windows Systems

Click Start, choose Run, and then enter the following:

ServerRoot/`stop-admin.cmd`

To Stop the Server From the Windows Control Panel

1. Open the Services control panel.

   Refer to Windows help for details on where to find the Services control panel.

2. Select Sun ONE Administration Server Version 5.2 from the list of services and then click Stop.

3. Click Close to exit the Services control panel.

NOTE	If you stop the Administration Server from the Windows Control Panel, you cannot restart it from within the Console. You must start the server from the command line or from the Windows Control Panel. For more information, see the preceding sections: “To Restart the Server From the Command Line” and “To Restart the Server From the Windows Control Panel.”

Logging Options

Log files can help you monitor activity on an instance of Administration Server, and can also help you troubleshoot server problems. Administration Server generates two kinds of logs:

Access log. Displays information about requests to the server and the responses from the server. By default, the file is located at `admin-serv/logs/access`.

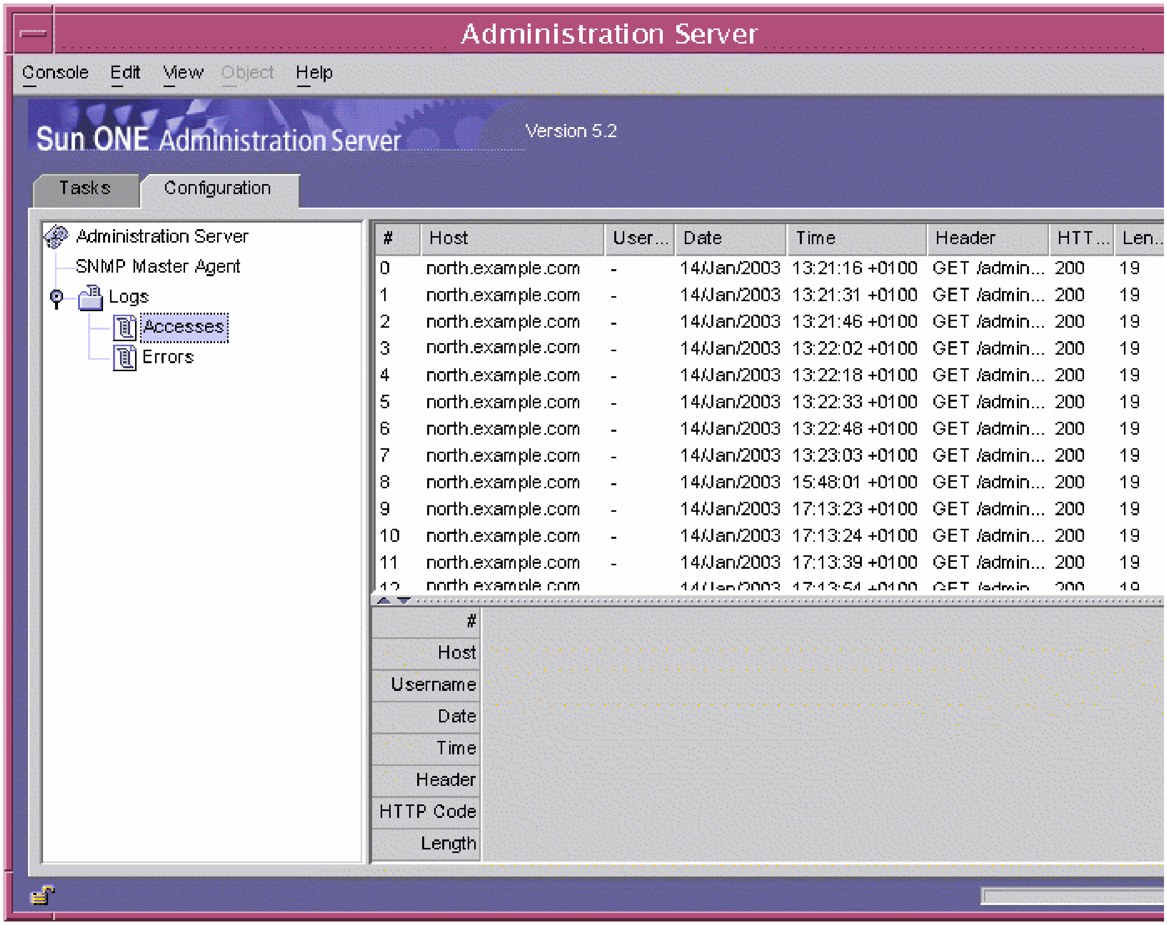
Error log. Displays errors the server has encountered since the log file was created. It also contains informational messages about the server, such as when the server was started and who tried unsuccessfully to log on to the server. By default, the file is located at `admin-serv/logs/error`.

You can view logs from Sun ONE Server Console. You can also change where logs are stored, for instance if you want Administration Server to write all log files to a shared folder.

To View the Access Log

1. From the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to view the access log for.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab.
4. In the configuration tree, expand the Logs directory, and then click the Accesses icon.

Figure 6-2 Administration Server Access Log



#	Host	User...	Date	Time	Header	HTT...	Len...
0	north.example.com	-	14/Jan/2003	13:21:16 +0100	GET /admin...	200	19
1	north.example.com	-	14/Jan/2003	13:21:31 +0100	GET /admin...	200	19
2	north.example.com	-	14/Jan/2003	13:21:46 +0100	GET /admin...	200	19
3	north.example.com	-	14/Jan/2003	13:22:02 +0100	GET /admin...	200	19
4	north.example.com	-	14/Jan/2003	13:22:18 +0100	GET /admin...	200	19
5	north.example.com	-	14/Jan/2003	13:22:33 +0100	GET /admin...	200	19
6	north.example.com	-	14/Jan/2003	13:22:48 +0100	GET /admin...	200	19
7	north.example.com	-	14/Jan/2003	13:23:03 +0100	GET /admin...	200	19
8	north.example.com	-	14/Jan/2003	15:48:01 +0100	GET /admin...	200	19
9	north.example.com	-	14/Jan/2003	17:13:23 +0100	GET /admin...	200	19
10	north.example.com	-	14/Jan/2003	17:13:24 +0100	GET /admin...	200	19
11	north.example.com	-	14/Jan/2003	17:13:39 +0100	GET /admin...	200	19
12	north.example.com		14/Jan/2003	17:13:54 +0100	GET /admin...	200	19

To View the Error Log

1. From the Sun ONE Server Console navigation tree, select the instance of Administration Server whose error log you want to view.

2. Click Open to open the management window for the instance of Administration Server.

3. Click the Configuration tab.

4. In the configuration tree, expand the Logs directory, then click the Errors icon.

To Change Where Logs Are Stored

1. From the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to modify.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Task tab, and then click Logging Options.
4. In the Logging Options window, enter new paths as necessary:

   Access Log - Log File. Enter a path to the directory where you want Administration Server to store the access log file. You can enter an absolute path or a path relative to your server root directory.

   Error Log - Log File. Enter the path to the directory where you want Administration Server to store the error log file. You can enter an absolute path or a path relative to your server root directory.
5. Click OK.
6. Restart the Administration Server for the changes to take effect.

Chapter 7

Administration Server Configuration

This chapter describes the configuration options you can use with Sun ONE Administration Server. It contains the following sections:

- Network Settings
- Access Settings
- Encryption Settings
- Directory Settings

In addition to the procedures described here, you may also configure settings as a quick task. Start by selecting the Administration Server Tasks tab, then click Configure Admin Server.

Network Settings

Network settings affect the way an instance of Sun ONE Administration Server runs. By default, these settings are configured automatically during installation, but you can modify them if your system configuration changes. You can change the following settings in the Console for the Administration Server in the fields on the Network tab page under the Configuration tab:

- Port Number
- IP Address
- Connection Restrictions

The *port number* specifies where an instance of Administration Server listens for messages. It can be any number between 1 and 65535 but, to avoid conflicts with other resources using reserved ports, it is typically a number greater than 1024 on UNIX systems. For security reasons, consider changing the port number regularly.

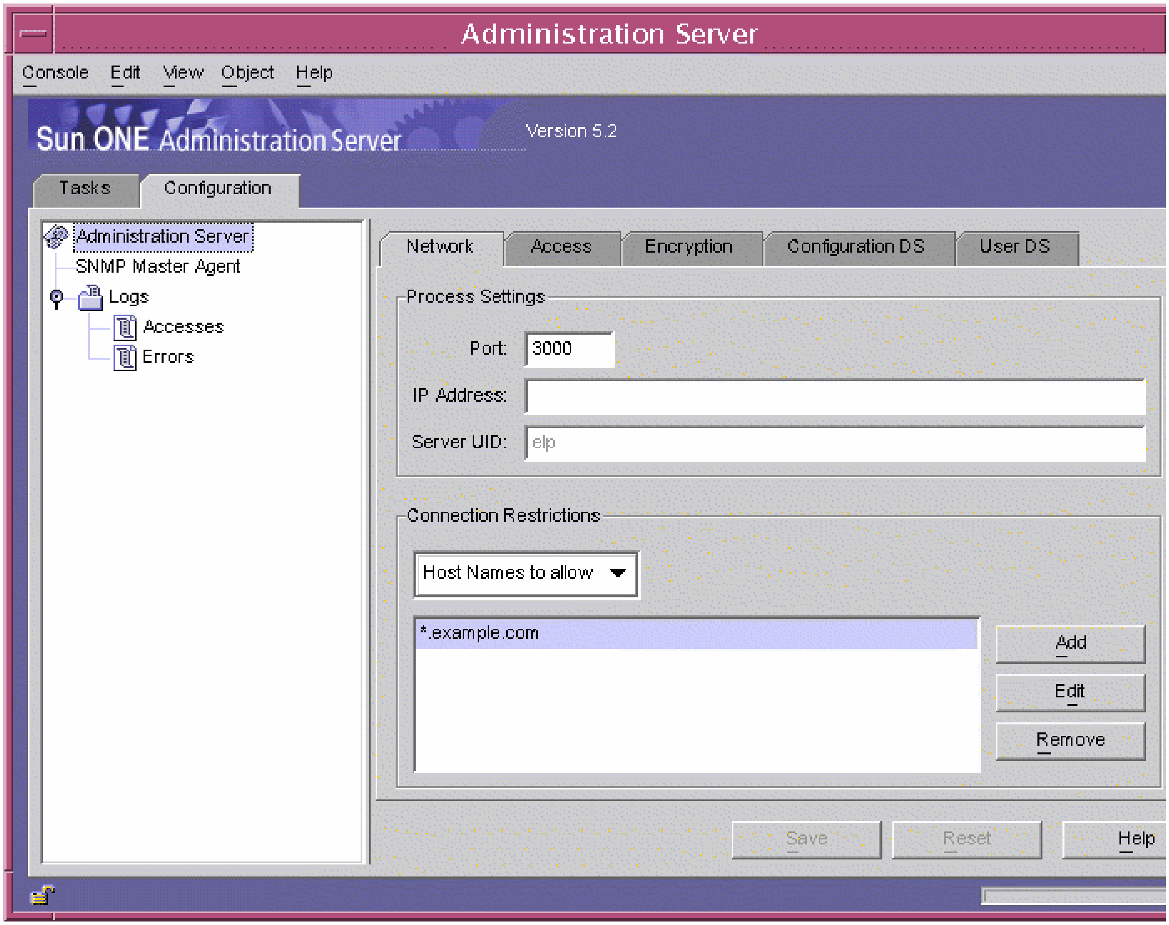
The *IP address* you specify is the one the Administration Server listens on for requests and messages. If you do not specify an IP address, the Administration Server listens for traffic on all IPv4 and IPv6 network interfaces available on the host system.

Connection restrictions allow you to specify which hosts are allowed to connect to an instance of Administration Server. You can list these hosts by DNS name, IP address, or both. You can use the `*` wildcard to specify a group of hosts. For instance, entering `*.example.com` allows all hosts in the `example.com` domain to access the instance. Entering `192.168.0.*.` allows all hosts whose IP addresses begin with `192.168.0` to access the instance. When specifying IPv4 address restrictions, you must include all three separating dots. If you do not, you receive an error message.

To Configure Network Settings

1. From the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to configure.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab, and then click the Network tab.

Figure 7-1 Configuring Network Settings



Process Settings

Port: 3000

IP Address:

Server UID: elp

Connection Restrictions

Host Names to allow

*.example.com

4. Enter network settings:

   Port. Enter the port number you want this instance of Administration Server to use. The port number can be any number between 1 and 65535. However, to avoid conflicts, the port number is typically a number greater than 1024.

   IP Address. (Optional) Enter the IP address on which to listen for traffic.

   Connection Restrictions. Displays a list of hosts allowed to connect to this instance of Administration Server. Use the drop-down list to specify whether you're adding to the list by DNS name or by IP address. The list is evaluated first by host names, and then by IP addresses.

   Add. Click if you want to display a dialog box for adding a host to the list of computers allowed to connect to this instance of Administration Server.

   Edit. Click if you want to display a dialog box for editing a Host IP address or DNS name on the list of computers allowed to connect to this instance Administration Server.

   Remove. Click if you want to remove a selected entry from the list of allowed hosts.

5. Click OK.

Access Settings

You can use the Access Settings tab to specify a user name and password for the Administration Server Administrator. The Administration Server Administrator is a special user that has full access to all features in an instance of Administration Server. This user is created during installation for the purpose of starting Sun ONE Server Console if a Directory Server is unavailable. The Administration Server Administrator user name and password are stored in the file *ServerRoot*/`admin-serv/config/admpw`.

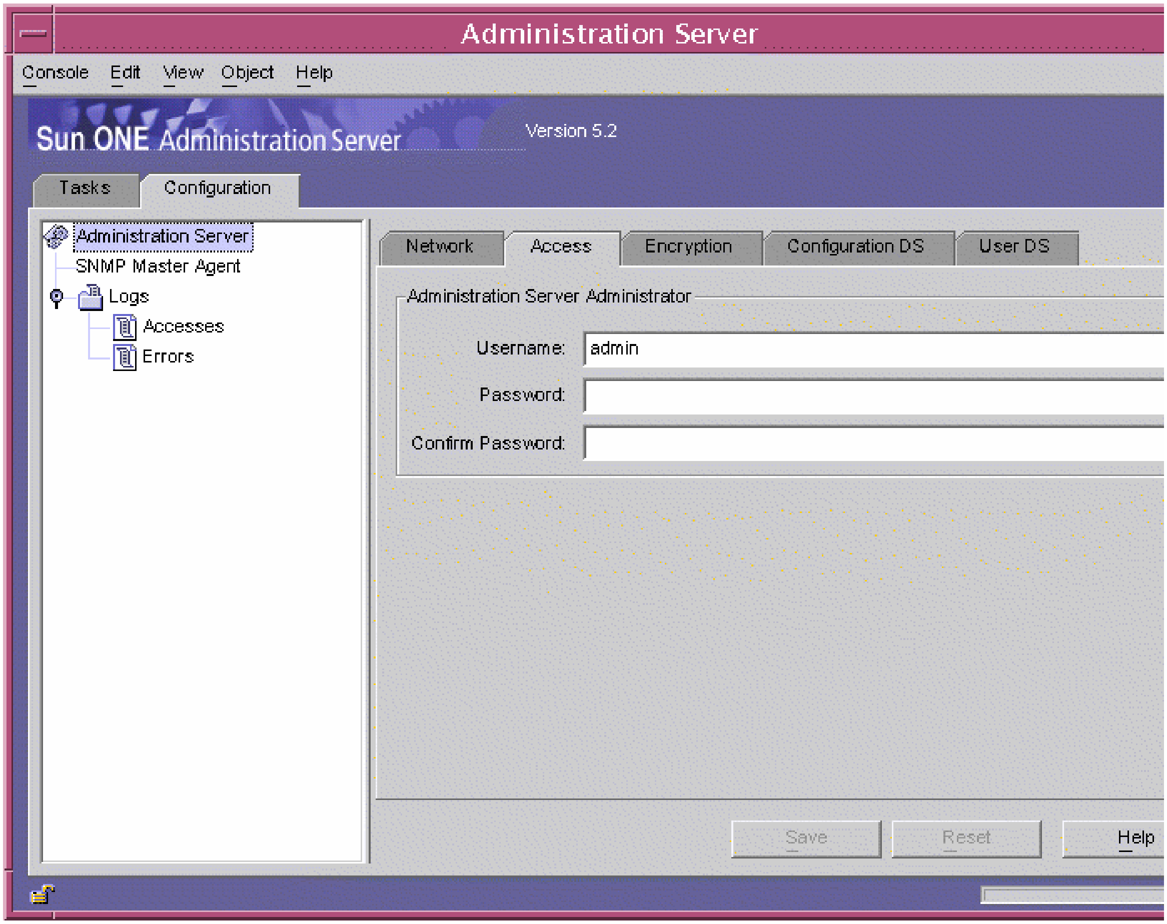
To Set Administration Server Access Settings

1. From the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to set Access Settings for.

2. Click Open to open the management window for the instance of Administration Server.

3. Click the Configuration tab, and then click the Access tab.

Figure 7-2 Configuring Access Settings



Administration Server Administrator

Username: admin

Password:

Confirm Password:

4. Enter access information:

   User name. Enter the user ID for the Administration Server Administrator.

   Password. Enter the Administration Server Administrator’s password.

   Confirm Password. Enter the password again to confirm it.

5. Click OK.

Encryption Settings

All Sun ONE servers support the Secure Sockets Layer (SSL) protocol and PKCS #11 APIs for encryption communication. Encryption protects communication between Administration Server and other servers from eavesdropping and tampering. You need to configure Administration Server for SSL if it communicates with SSL-enabled servers.

Before you can use SSL with Administration Server, you must first request and install a certificate, and then activate SSL on the server. The following procedures walk you through requesting and installing a certificate, as well as activating SSL on an instance of Administration Server.

To Request and Install a Certificate for Administration Server

1. In the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to install a certificate on.
2. Click Open to open the management window for the instance of Administration Server.
3. In the Administration Server management window, open the Console menu, and choose Security > Manage Certificates.
4. Click the Request button, and then provide information as prompted.

   See "Obtaining and Installing a Server Certificate," on page 145 for detailed information.
5. Once you have a certificate, click the Install button, and then provide information as prompted.

Once you've installed a certificate, activate SSL as described in the next procedure.

To Activate SSL on Administration Server

1. In the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to activate SSL encryption on.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab.

4. Click the Encryption tab.
5. Select "Enable SSL for this server."

   The options in the following steps become available only when you turn on SSL encryption.
6. Select "Use this cipher family: RSA."
7. Choose the security device where your key is stored:

   If the key is stored in the local key database, select "Internal (Software-based)." If the key is stored on an external device (such as a SmartCard), select that device.
8. Choose the certificate you want to use with SSL.

   Certificate information is stored in the certificate database. If you're not sure which certificate to use, view the Certificate Management dialog box for more information. To view the Certificate Management dialog box, from the File menu, choose Certificate Management.
9. Click the Settings button.
10. Set the ciphers that this instance of Administration Server should accept when communicating securely with Sun ONE Server Console, or other servers.

    First, click a tab for a version of SSL or TLS. Then, choose the ciphers that you want this instance of Administration Server to accept when communicating over that version of SSL or TLS.
11. Select Require Client Authentication if you want the server to authenticate the client during the SSL handshake.
12. Click Save.

Directory Settings

Directory settings tell the Administration Server where to find the configuration directory and the user directory.

The Configuration Directory

When you install an Sun ONE server, you are prompted for the location of an instance of Directory Server in which to store configuration data. Depending on the way your organization uses directories, you specify either an instance of Directory Server that contains only configuration data or an instance of Directory Server that contains both user and configuration data.

Configuration data is stored under `o=NetscapeRoot` in the instance of Directory Server that you specify during installation. This subtree is called the configuration directory and contains server settings such as network topology information and server instance entries. When you install a server or change its configuration, the new settings are stored in the configuration directory subtree.

Changing the Host or Port Number

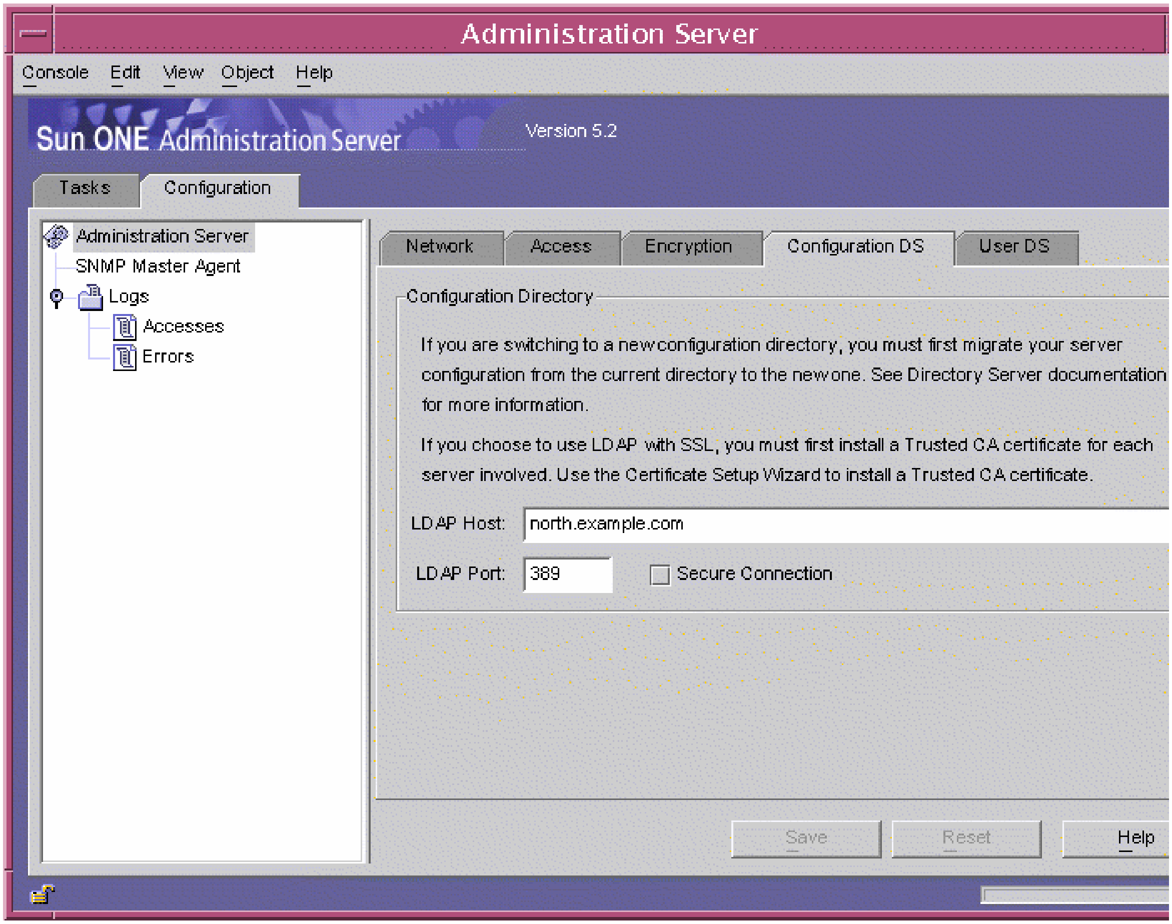
You can designate a different host or port number for the instance of Directory Server containing the configuration directory subtree.

CAUTION Changing the Directory Server host name or port number affects the rest of the servers in the server group. If you change a setting here, you must make the same change in every server in the server group.

To Change the Host or Port Number

1. In the Sun ONE Server Console navigation tree, select an instance of Administration Server. This is the server for which you may change the configuration directory.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab.
4. Click the Configuration DS tab.

Figure 7-3 Configuration Directory Settings



Configuration Directory

If you are switching to a new configuration directory, you must first migrate your server configuration from the current directory to the new one. See Directory Server documentation for more information.

If you choose to use LDAP with SSL, you must first install a Trusted CA certificate for each server involved. Use the Certificate Setup Wizard to install a Trusted CA certificate.

LDAP Host: north.example.com

LDAP Port: 389

□ Secure Connection

5. Modify settings as appropriate:

   LDAP Host. Enter the host name of the configuration Directory Server this instance of Administration Server uses.

   LDAP Port. Enter the port number for the configuration Directory Server this instance of Administration Server uses.

   Secure Connection. Check this box if you want to connect securely with the configuration Directory Server. Before choosing this option, make sure the configuration Directory Server running on the specified LDAP Host and LDAP Port already has SSL activated on it.

6. Click Save.

The User Directory

The user directory is stored in a Directory Server subtree that you create. The user directory is used for authentication, user management, and access control. It stores all user and group data, account data, group lists, and access control instructions (ACIs).

You can have more than one user directory in your enterprise. For example, to increase directory performance, one company might deploy three user directories, one in each of three geographic regions. Another company might deploy five user directories, one for each of five Mail Servers. You can configure an instance of Administration Server to authenticate users against multiple user directories.

If the user and configuration directory subtrees are in different instances of Directory Server, you need to activate pass-through authentication.

User Directory Settings

When you're installing an Sun ONE server, you are prompted to specify a user directory that is associated with the administration domain in which the server is located. By default, this association is inherited at all levels beneath the administration domain. Server groups and the individual servers within them use the same user directory as the domain.

There may be times when you need to override default user directory settings at the server group or domain level. For example, you may need to change the user directory for a domain when you upgrade to a new Directory Server. Alternatively, you might want to temporarily change the user directory for a server group when you are testing a new instance of Directory Server and do not want to use your existing user directory with it.

User Authentication and Directory Failover Support

When a user logs in to Sun ONE Server Console, the user enters a user ID that is checked against the user directory. If the user ID cannot be authenticated in a user directory, the user cannot successfully log in to Sun ONE Server Console.

You can employ more than one user directory for authenticating user IDs. This is useful when the instance of Directory Server containing your primary user directory is not accessible. If the user directory has been replicated on other hosts, Sun ONE Server Console continues to check the user ID against each user directory in the list until authentication succeeds or there are no more entries in the list. This ability to check multiple instances of Directory Server is called *failover support*.

To list the user directories to use for failover support, follow the instructions for "To Change the User Directory Settings for a Domain," on page 109 or "To Change User Directory Settings for a Server Group," on page 111.

Changing User Directory Settings for a Domain

If you are the configuration administrator, you can change the user directory settings for a domain.

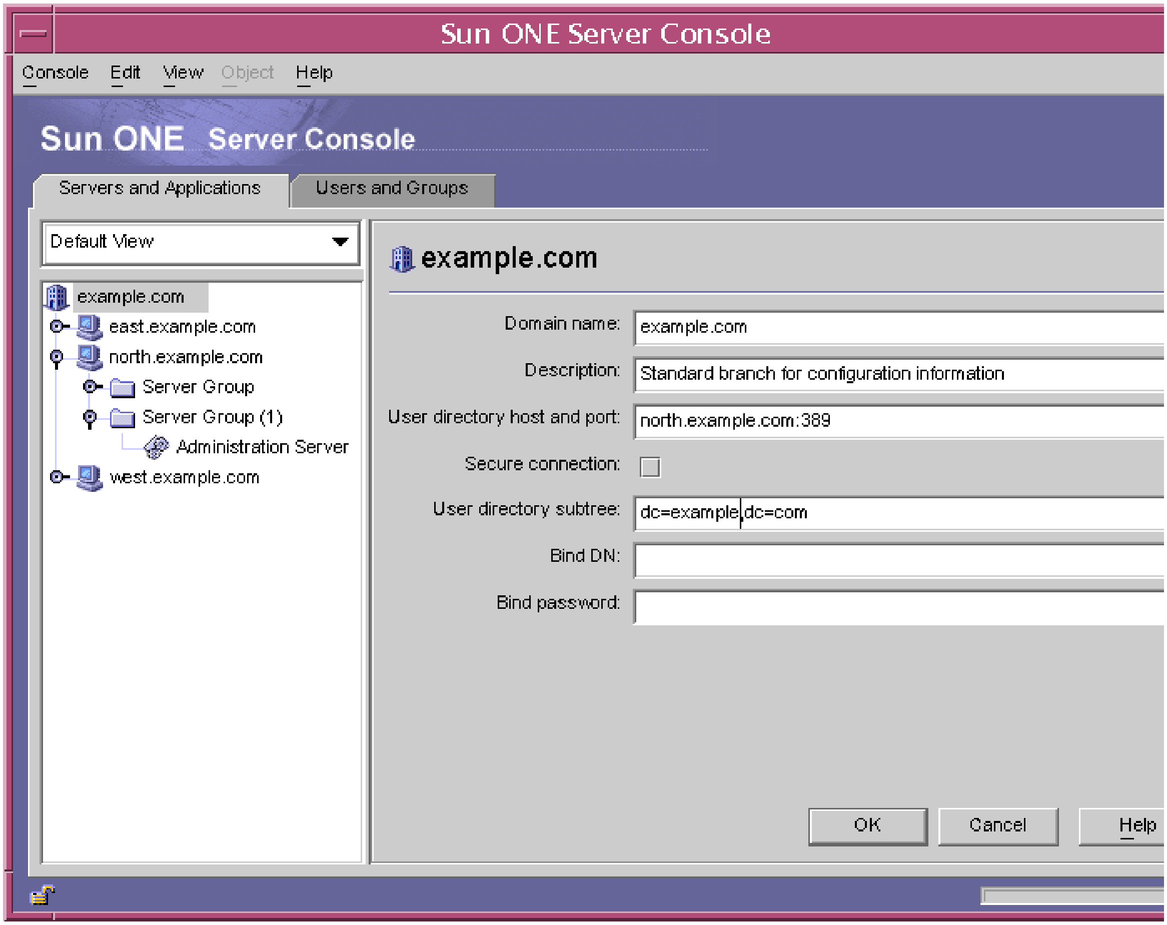
CAUTION Changing the Directory Server host name or port number affects the rest of the servers in the administration domain. If you change a setting here, you must restart all the servers in the administration domain.

To Change the User Directory Settings for a Domain

1. In the Sun ONE Server Console navigation tree, select the administration domain that you want to change user directory settings for.

2. In the right-hand panel of the main Sun ONE Server Console window, click Edit.

Figure 7-4 User Directory Settings



Domain name: example.com

Description: Standard branch for configuration information

User directory host and port: north.example.com:389

Secure connection: □

User directory subtree: dc=example,dc=com

Bind DN:

Bind password:

3. Modify domain information as appropriate.

   Domain name. Enter a domain name. Example: `example.com`

   Description. Enter a name that helps you identify this domain.

   User directory host and port. Specify the location of the new user directory using the host computer's fully qualified domain name and port number. For authentication purposes, you can enter more than one user directory location separated by spaces. Example:

   ```
   eastcoast.example.com:389 westcoast.example.com:389
   ```

   See "User Authentication and Directory Failover Support" on page 108 for more information.

   If you specified more than one location in the "User directory host and port" field, the settings for the remaining fields apply to them all.

   Secure connection. Check this box if you want to connect securely with the user directory. Before choosing this option, make sure the instance of Directory Server is running on the specified user directory host and port already has SSL activated on it.

   User directory subtree. Enter the location of the new user directory. Example: `dc=example,dc=com`

   This subtree must contain the user directory in all the locations specified in the "User directory host and port" field.

   Bind DN. (Optional) Enter the distinguished name for a user who can access the new user directory. Example: `uid=john, ou=people, dc=example,dc=com`.

   Bind password. (Optional) Enter the password of the user specified by the Bind DN.

4. Click Save.

To Change User Directory Settings for a Server Group

1. In the Sun ONE Server Console navigation tree, expand the server group that you want to change user directory settings for.

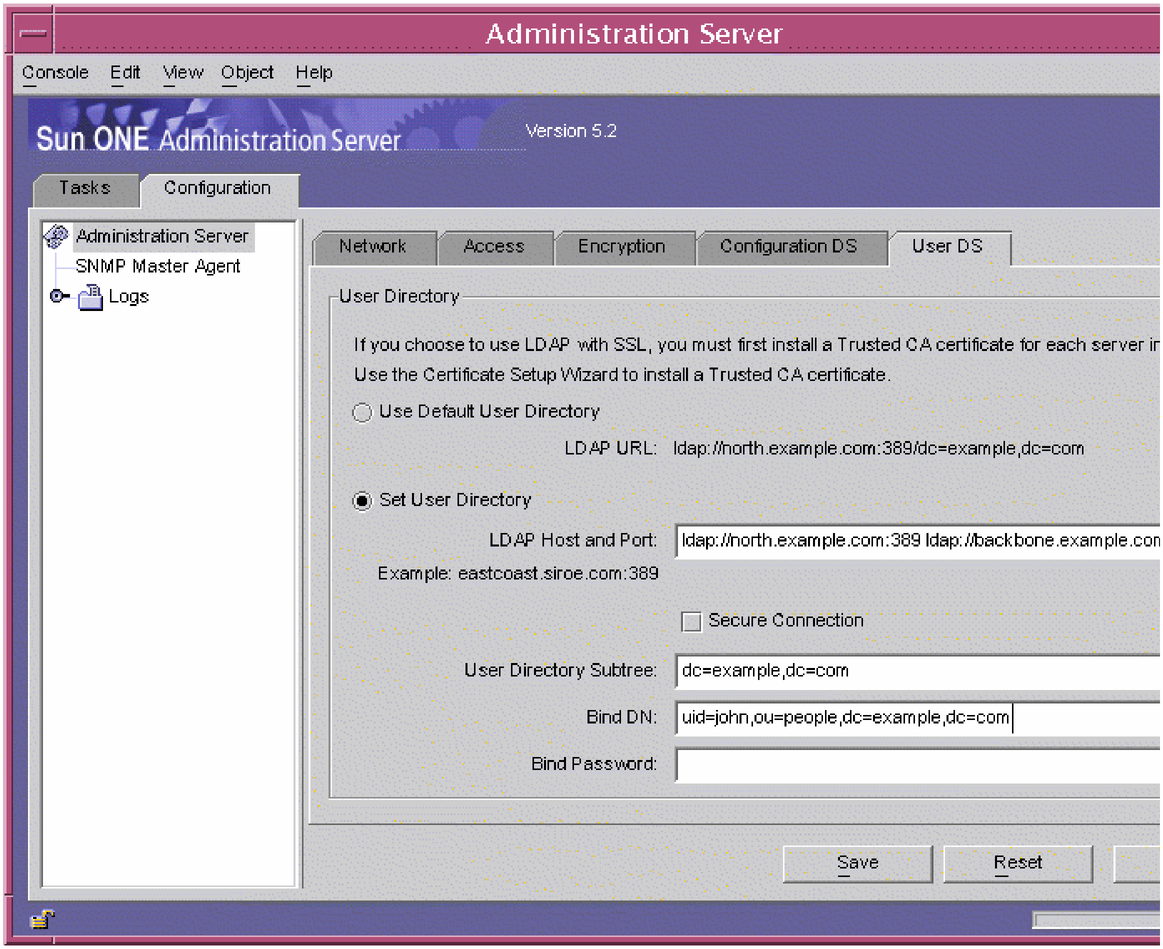
2. Select the instance of Administration Server in the server group.

3. Click Open to open the management window for the instance of Administration Server.

4. Click the Configuration tab.

5. Click the User DS tab.

Figure 7-5 User Directory Settings For a Group



User Directory

If you choose to use LDAP with SSL, you must first install a Trusted CA certificate for each server ir
Use the Certificate Setup Wizard to install a Trusted CA certificate.

○ Use Default User Directory

LDAP URL: ldap://north.example.com:389/dc=example,dc=com

● Set User Directory

LDAP Host and Port: ldap://north.example.com:389 ldap://backbone.example.con
Example: eastcoast.siroe.com:389

□ Secure Connection

User Directory Subtree: dc=example,dc=com

Bind DN: uid=john,ou=people,dc=example,dc=com

Bind Password:

6. Modify settings as appropriate.

   Use Default User Directory. Select this option if you want to use the default user directory associated with the domain.

   Set User Directory. Select this option if you want to use a user directory other than the default associated with the domain.

   LDAP Host and Port. Specify the location of the user directory using the host computer's fully qualified domain name and port number. For authentication purposes, you can enter more than one user directory location separated by spaces. Example:

   ```
   eastcoast.example.com:389 westcoast.example.com:389
   ```

   See "User Authentication and Directory Failover Support" on page 108 for more information.

   If you specified more than one location in the "LDAP Host and Port" field, the settings for the remaining fields apply to them all.

   Secure Connection. Check this box if you want to connect securely with the user directory. Before choosing this option, make sure the instance of Directory Server is running on the specified user directory host and port already has SSL activated on it.

   User Directory Subtree. Enter the location of the new user directory. Example: `dc=example,dc=com`

   This subtree must contain the user directory in all the locations specified in the "LDAP Host and Port" field.

   Bind DN. (Optional) Enter the distinguished name for a user who can access the new user directory. Example: `uid=john, ou=people, dc=example,dc=com`.

   Bind Password. (Optional) Enter the password of the user specified by the Bind DN.

7. Click Save.

Chapter 8

Administration Server Command-Line Tools

The command-line tools described in this chapter come with Sun ONE Administration Server. You can use these utilities to configure an instance of Administration Server without launching Sun ONE Server Console:

- mpsadmconfig
- mpsadmserver admin_ip
- ldapsearch, ldapmodify, and ldapdelete

This chapter tells you how to use the command-line tools.

mpsadmconfig

The `mpsadmconfig`(1M) utility allows you to configure an instance of Administration Server using the command line instead of using the Sun ONE Server Console. When using the product delivered in Solaris package format, use `mpsadmconfig`(1M) to modify network, access, encryption, or directory settings.

When using the product not delivered in Solaris package format, use the `admconfig` utility stored under *ServerRoot*`/bin/admin` and has the same syntax as `mpsadmconfig`(1M).

Syntax for mpsadmconfig

`/mpsadmconfig` *[options] task [args] [task2] [args] [task3] [args] ...*

The options that you can use with `mpsadmconfig`(1M) are described in the section that follows. The tasks that you can perform with `mpsadmconfig`(1M), as well as the arguments for those tasks, are described in “Tasks and Their Arguments,” on page 117.

Options

An option is a general setting that affects how `mpsadmconfig`(1M) runs. You can specify an option using a complete command such as `-user` or an abbreviated command such as `-u`. When specifying a command, make sure to use enough characters to differentiate it from other commands.

Option commands are not case sensitive. For example, both `-USER` and `-User` are accepted as the `-user` command. You can use multiple option commands with the same invocation of `mpsadmconfig`(1M). For example, the following option commands specify that `mpsadmconfig`(1M) should establish an encrypted connection with `eastcoast.example.com` on port 1389.

```
/usr/sbin/mpsadmconfig -enc -ser eastcoast.example.com:1389
```

Table 8-1 Options You Can Use With `mpsadmconfig`(1M)

Commands for Options	What the Command Does
`-con[tinueOnError]`	Finishes any remaining tasks (that have been specified on the command line) when an error occurs. (Default behavior when any task fails is to quit without running the remaining tasks.)
`-enc[ryption]`	Uses encrypted HTTP (HTTPS) to connect to the server. (The default protocol is HTTP.)
`-h[elp]` *[task]*	Displays general usage information. Include a task name for usage information specific to that task.
`-i[nputFile]` *filename*	Reads options and tasks from the specified file. You can specify additional options on the command line. If an option is present on the command line and in the specified file, the command-line settings are used. If the `-inputFile` option is present in the specified file, it is ignored to prevent `admconfig` from reading multiple sets of options.
`-ser[ver]` *[host]:port*	Connects to the server on the specified host and port. If a host is not specified, the local host is used. The server port number (preceded by the colon) is required.

Table 8-1 Options You Can Use With `mpsadmconfig`(1M) *(Continued)*

Commands for Options	What the Command Does
`-u[ser]` *[uid]:[pwd]*	Connects to the server using the specified user name and password. If a user name is not specified, you are prompted for the current user's password. The password appears onscreen when it is typed, so if security is a concern, use the `-inputFile` option and list the user name and password in a file with suitable permissions. Note that if the `-user` option is specified, then, at a minimum, the colon must be specified. If the `-user` option is not specified, then the user is prompted for both the user name and password.
`-verb[ose] [0-9]`	Sets the level of screen output (9=full output, 0=no output).The default level is 9.
`-vers[ion]`	Displays the version and copyright information.

Tasks and Their Arguments

A task specifies an operation that `mpsadmconfig`(1M) should perform. Some tasks take arguments, commands that provide information necessary to complete an operation.

You can specify a task using a complete command such as `-restart` or an abbreviated command such as `-r`. When specifying a task command, make sure to use enough characters to differentiate it from other commands. The task commands are not case sensitive. Both `-RESTART` and `-Restart` are accepted as the `-restart` task.

You can run multiple tasks with the same invocation of `mpsadmconfig`(1M). If you use the `-i[nputFile]` option command to specify an input file, `mpsadmconfig`(1M) runs the tasks contained in that file first. The `mpsadmconfig`(1M) utility executes tasks in the order that they are specified in the input file and then in the order specified on the command line.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M)

Commands for Tasks	What the Command Does
`-countA[ccessLogEntries]`	Counts the number of entries in the access log file. Run this task before `-viewAccesslogEntries` to determine the number of entries in the access log.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M) *(Continued)*

Commands for Tasks	What the Command Does
`-viewA[cessLogEntries]`	Lets you view the specified entries in the error log file. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-viewAcessLogEntries \'`*start stop*`\'` **Required Arguments** *start* The number of the first log entry to display. *stop* The number of the last log entry to display. On UNIX systems, the backslash character is required before the quotes surrounding the *start* and *stop* arguments. If the backslash is not provided, the shell evaluates the quotes and pass the arguments without quotes to the command line. As a result, only *start* is assigned as a parameter for `-viewAcessLogEntries`, causing the operation to fail.
`-countE[rrorLogEntries]`	Counts the number of entries in the error log file. Run this task prior to `-viewErrorLogEntries` to determine the number of entries in the error log.
`-viewE[rrorLogEntries]`	Lets you view the specified entries in the error log file. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-viewErrorLogEntries \'`*start stop*`\'` **Required Arguments** *start* The number of the first log entry to display. *stop* The number of the last log entry to display. On UNIX systems, the backslash character is required before the quotes surrounding the *start* and *stop* arguments. If the backslash is not provided, the shell evaluates the quotes and pass the arguments without quotes to the command line. As a result, only *start* is assigned as a parameter for `-viewErrorLogEntries`, causing the operation to fail.
`-getAc[cessLog]`	Retrieves the path for the access log file for this instance of Administration Server.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M) *(Continued)*

Commands for Tasks	What the Command Does
`-setAc[cessLog]`	Specifies the path for the access log file for this instance of Administration Server. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setAccessLog` *filename* **Required Argument** *filename* Full path of the new server access log file.
`-getAdd[resses]`	Lets you view the IP addresses from which connections are allowed.
`-setAdd[resses]`	Specifies the IP addresses from which connections are allowed. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setAddresses` *addresses* **Required Argument** *addresses* New IP addresses and host names (separated by spaces) from which connections are allowed.
`-getAdminUI[D]`	Retrieves the Administration Server Administrator’s user name.
`-setAdminUI[D]`	Specifies the Administration Server Administrator’s user name. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setAdminUID` *uid* **Required Argument** *uid* The new Administration Server Administrator’s user ID.
`-setAdminP[wd]`	Specifies the Administration Server Administrator’s password. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setAdminPwd` *password* **Required Argument** *password* The new password for the Administration Server Administrator.
`-getAdminUs[ers]`	Retrieves the path of the `adminusers` file.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M) *(Continued)*

Commands for Tasks	**What the Command Does**
`-setAdminUs[ers]`	Specifies the path of the `adminusers` file. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setAdminUsers` *adminusers* **Required Argument** *adminusers* New path for the `adminusers` file.
`-getCa[cheLifetime]`	Displays the amount of time for which a user authentication is cached.
`-setCa[cheLifetime]`	Specifies the amount of time to cache a user authentication. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setCacheLifetime` *msec* **Required Argument** *msec* New cache lifetime in milleseconds.
`-getCl[assname]`	Retrieves the Java classname for this instance of Administration Server.
`-setCl[assname]`	Specifies the Java classname for this instance of Administration Server.
`-getDe[faultAcceptLanguage]`	Displays the default language for this instance of Administration Server.
`-setDe[faultAcceptLanguage]`	Specifies the default language for this instance of Administration Server. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setDefaultAcceptLanguage` *language* **Required Argument** *language* New default language. This is specified with an ISO 639 two letter code. For example, English is `en`.
`-getDS[Config]`	Retrieves the current LDAP server host, port, and base DN, and identifies whether the LDAP server is running SSL.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M) *(Continued)*

Commands for Tasks	What the Command Does
`-setDS[Config]`	Specifies the LDAP server host, port, and base DN, and specifies whether the LDAP server is running SSL. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setDSConfig \'`*host port baseDN ssl*`\'` **Required Arguments** *host* The LDAP Server host name. *port* The LDAP Server port number. *baseDN* The LDAP Server base DN. *ssl* Specify `true` or `false` depending on whether the LDAP server is already using the Secure Sockets Layer (SSL) protocol to communicate with this instance of Administration Server. On UNIX systems, the backslash character is required before the quotes surrounding the `these` arguments. If the backslash is not provided, the shell evaluates the quotes and pass the arguments without the quotes to the command line. As a result, only *host* is assigned as a parameter for `-setDSConfig`, causing the operation to fail.
`-getU[GDSConfig]`	Retrieves the current user and group LDAP server information, including the host, port, base DN, and authentication DN.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M) *(Continued)*

Commands for Tasks	What the Command Does
`-setU[GDSConfig]`	Specifies the host, port, base DN, authentication DN, and authentication password for the instance of Directory Server containing the user and group directory. You can invoke `-setUGDSConfig` either with or without arguments. If you invoke this task without any arguments, the Directory Server configuration is reset to the installation defaults. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setUGDSConfig` [`\'`*host port baseDN ssl uid pwd*`\'`] **Optional Arguments** If you want to override the current user and group settings, you must provide all six of the following arguments: • *host* The host name on which the instance of Directory Server is running. • *port* The port number on which the instance of Directory Server is running. • *baseDN* The base DN for the instance of Directory Server. • *ssl* Specify `true` or `false` depending on whether the instance of Directory Server is already using the Secure Sockets Layer (SSL) protocol to communicate with this instance of Administration Server. • *uid* The Distinguished Name used to bind to the instance of Directory Server. Example: `dn: uid=dfauvarque, ou=people, dc=example, dc=com` • *pwd* The password used to bind to the instance of Directory Server. On UNIX systems, the backslash character is required before the quotes surrounding these arguments. If the backslash is not provided, the shell evaluates the quotes and pass the arguments without quotes to the command line. As a result, only *host* is assigned as a parameter for `-setUGDSConfig,` causing the operation to fail. The *host*, *port*, *baseDN*, and *ssl* arguments are used to create the LDAP URL for the `ugdsconfig.dirurl` attribute. The *uid* argument is used to set the `ugdsconfig.binddn` attribute, and the *pwd* argument is used to set the `ugdsconfig.bindpw` attribute.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M) *(Continued)*

Commands for Tasks	What the Command Does
`-setU[GDSConfig]` (continued)	Note that the space character is used to parse these six arguments. Therefore, none of the arguments can have spaces in them. To indicate spaces within an argument, use the + character. For example, to specify `cn=directory manager` as the value for the *uid* attribute, enter `cn=directory+manager`. Since the + character is used in place of the space character, you cannot use it as an actual value.
`-getE[rrorLog]`	Retrieves the path for the server error log file.
`-setE[rrorLog]`	Specifies the path for the server error log file. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setErrorLog` *filename* **Required Argument** *filename* Full path of the new server access log file.
`-getH[osts]`	Lets you view the host names from which connections are allowed.
`-set[Hosts]`	Specifies the host names from which connections are allowed. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setHosts` *hosts* **Required Argument** *hosts* host names from which connections are allowed.
`-getO[neACLDir]`	Retrieves the path for the ACL folder.
`-setO[neACLDir]`	Specifies the path for the ACL folder. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setOneACLDir` *directory* **Required Argument** *directory* Path for the ACL folder.
`-getPo[rt]`	Lets you view the port number that this instance of Administration Server is using.

Table 8-2 Tasks You Can Perform With `mpsadmconfig`(1M) *(Continued)*

Commands for Tasks	What the Command Does
`-setPo[rt]`	Specifies the port number that this instance of Administration Server should use. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setPort` *port* **Required Argument** *port* Port number that this instance of Administration Server should use.
`-getSe[rverAddress]`	Retrieves the IP address of this instance of Administration Server.
`-setSe[rverAddress]`	Specifies the IP address that this instance of Administration Server should use. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setServerAddress` *address* **Required Argument** *address* IP address that this server should use.
`-getSy[stemUser]`	Retrieves the user name that this instance of Administration Server runs as.
`-setSy[stemUser]`	Specifies the user name that this instance of Administration Server should run as. **Syntax** `/usr/sbin/mpsadmconfig` *[options]* `-setSuiteSpotUser` *user* **Required Argument** *user* User ID that this instance should run as.
`-r[estart]`	Restarts this instance of Administration Server.
`-st[op]`	Stops this instance of Administration Server.

Examples

The following examples demonstrate different uses of `admconfig`.

- This example changes the port number for an instance of Administration Server to 33333, and then restarts the instance. The verbose level option, which controls how much status information is printed to the screen, is set to 5.

  ```
  /usr/sbin/mpsadmconfig -server eastcoast.example.com:22222 -user
  josu:password -verbose 5 -setPort 33333 -restart
  ```

- This example retrieves the hosts from which connections are allowed. The verbose level option is set to 9 (the default value when a number isn't specified).

  ```
  /usr/sbin/mpsadmconfig -ser eastcoast.example.com:33333 -u
  josu:password -verb -geth
  ```

- This example displays the help information for restarting an instance of Administration Server.

  ```
  /usr/sbin/mpsadmconfig -h r
  ```

mpsadmserver admin_ip

When your computer system's IP address changes, you must update the local Administration Server configuration file and the configuration directory. If you do not enter the new IP address in these locations, you cannot subsequently start the Administration Server.

When using the product delivered in Solaris package format, use the `mpsadmserver`(1M) utility with the `admin_ip` subcommand to update these two configurations.

When using the product not delivered in Solaris package format, a Perl script is provided to help you update these two configurations. The script changes the IP address for an instance of Administration Server in both the `local.conf` file and the configuration directory. The script is called `admin_ip.pl` and is stored in the *ServerRoot*`/shared/bin` folder.

Usage

To run the command follow the instructions for the appropriate platform:

On UNIX Systems

Enter the command appropriate for the version you installed, such as the following for Solaris:

```
/usr/sbin/mpsadmconfig admin_ip Directory_Manager_DN
Directory_Manager_password old_IP new_IP [port #]
```

The old IP address is saved in a file called `local.conf.old`.

On Windows Systems

From the command line go to the *serverRoot*/`shared/bin` folder and enter

```
../../install/perl admin_ip.pl Directory_Manager_DN
Directory_Manager_password old_IP new_IP [port #]
```

The old IP address is saved in a file called `local.conf.old`.

ldapsearch, ldapmodify, and ldapdelete

These tools allow you to search and modify the user directory. These tools are documented in Solaris online manual pages `ldapsearch`(1), `ldapmodify`(1), and `ldapdelete`(1). Expanded versions are provided with the Sun ONE Directory Server Resource Kit and documented in the *Sun ONE Directory Server Resource Kit Tools Reference*.

Part 4

Advanced Server Management

Chapter 9

Access Control

This chapter describes how you can use access control instructions to define who can manage and use Sun ONE servers. It contains the following sections:

- Overview of Access Control
- Setting Access Permissions For Servers
- Working With Access Control Instructions

Overview of Access Control

If a number of administrators in your enterprise use Sun ONE Server Console, you may want to restrict what each of them can see and do. For example, you may want one administrator to handle all server management tasks and another to manage users and groups. You can specify these permissions through the use of *Access Control Instructions* (ACIs).

ACIs are rules that permit or restrict access to a server, on screen element, task, or directory entry. In a single ACI, you can specify access based on user name, IP address, time of day, and a number of other criteria. You can also chain multiple ACIs together in an *Access Control List* (ACL) to perform complex authorization procedures.

For users, access control is transparent. During login, Sun ONE Administration Server authenticates a user against Directory Server. Directory Server returns the user's administrative privileges and applicable ACIs. The instance of Administration Server evaluates this information and then instructs Sun ONE Server Console to display only those resources and server tasks that the user is allowed to access.

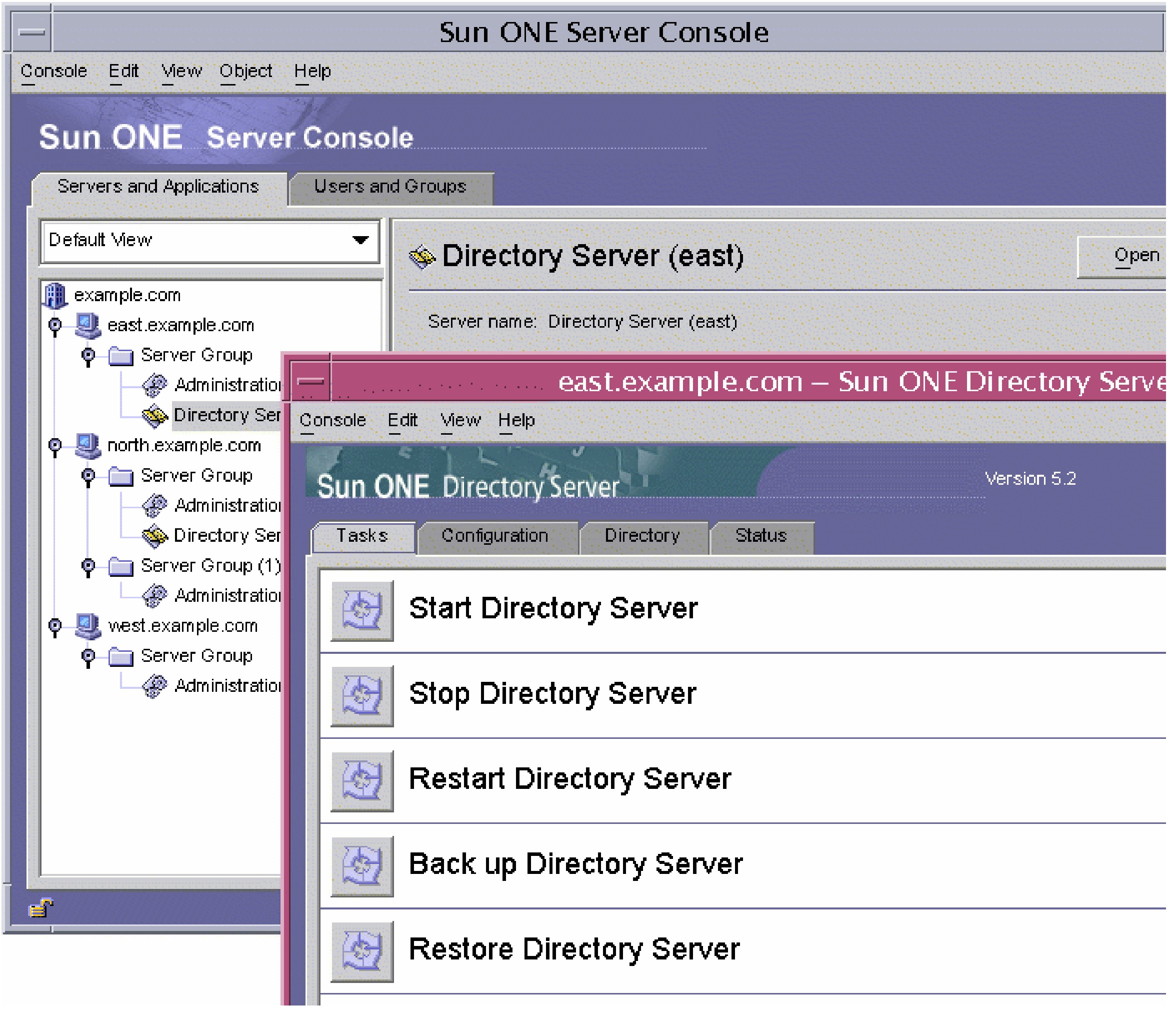
For detailed information about ACIs for a particular Sun ONE server, see the documentation for that server.

Examples of Access Control

The following examples illustrate how an organization might use ACIs to grant and restrict access to servers and data by different administrators.

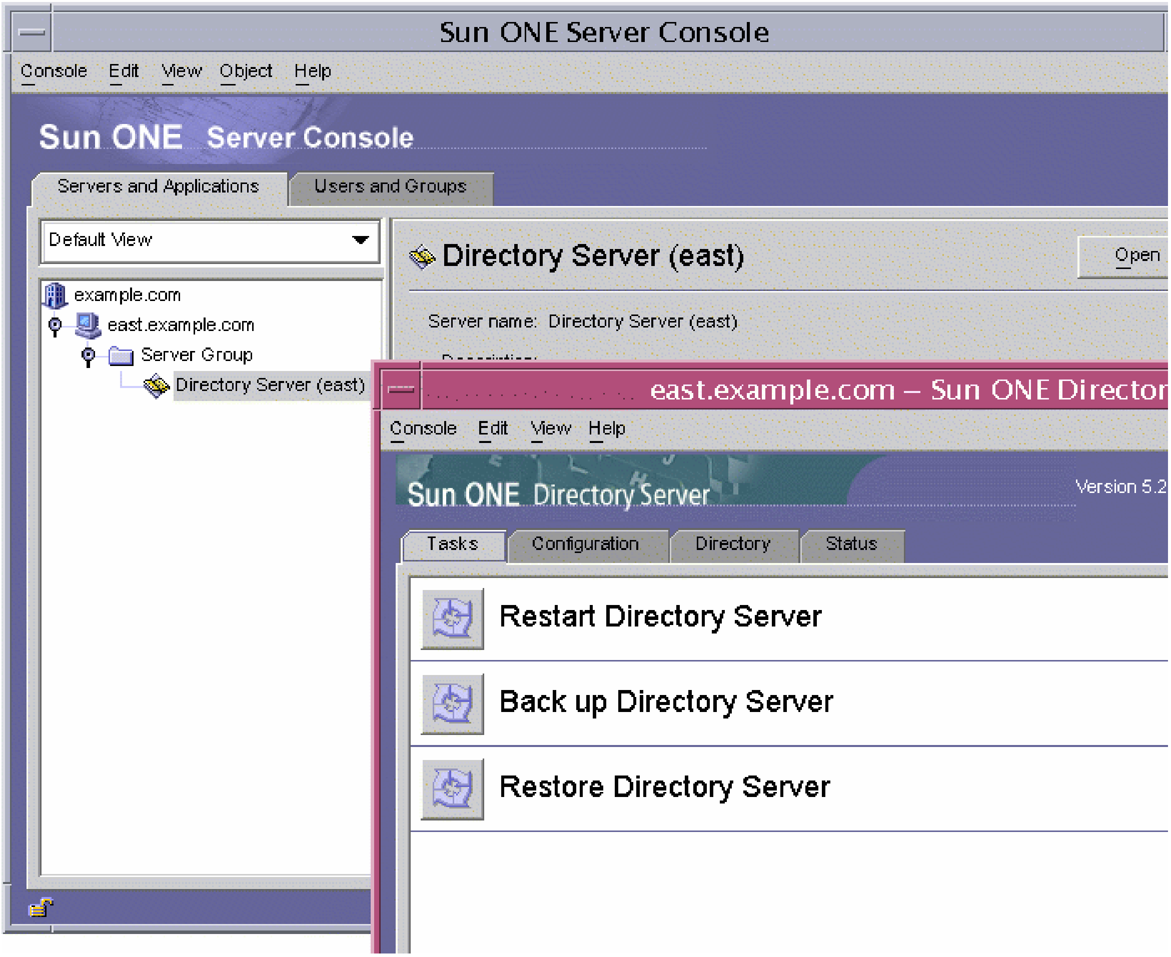
Jane is an administrator who troubleshoots network problems. She needs to be able to access any server in the enterprise and frequently modifies user account information. As a result, the Configuration Administrator has placed very few restrictions on what she can access. When Jane logs into Sun ONE Server Console, she has a complete view of servers, tabs, and tasks.

Figure 9-1 Unrestricted View of Resources and Tasks



John is also an administrator, but his job is focused on managing instances of Directory Server in the enterprise. As a result, the Configuration Administrator has used ACIs to restrict the onscreen elements and tasks that he can access. When John logs into Sun ONE Server Console, he sees only the servers and tasks required to do his job.

Figure 9-2 Restricted View of Resources and Tasks



Server name: Directory Server (east)

Setting Access Permissions For Servers

You can specify which users have administrative access to servers in the Sun ONE Server Console navigation tree by using the Set Permissions dialog box.

To Set Access Permissions for a Server in the Navigation Tree

1. Select a server in the Sun ONE Server Console navigation tree.

2. From the Object menu, choose Set Access Permissions.

   Alternatively, you can right-click, and then choose Set Access Permissions.

3. In the Set Permissions Dialog window, specify who has administrative access to the server.

   To add a user to the list of people who can administer the server, click the Add button, and then search for the user or group that you want to grant administrative rights to. For more information on locating users and groups in the directory, see "Locating a User or Group in the Directory," on page 63.

   To remove a user from the list, select the user, and then click the Delete button.

   Note that granting a user the right to administer a server does not automatically allow that user to give others the same right. If you want to allow a user to grant administrative rights to other users, you must add him or her to the Configuration Administrators group. For instructions on how to do this, see "To Add Users to the Configuration Administrators Group," on page 77.

4. Click OK when you have finished specifying who can access the server.

Working With Access Control Instructions

When you create Access Control Instructions (ACIs) you specify which users can manage a resource as well as when and how access is granted. Sun ONE Server Console uses two tools to simplify the process of creating and assigning ACIs: ACI Manager and ACI Editor.

The ACI Manager lets you apply ACIs to an object. It is also the dialog box from which you typically launch the ACI Editor.

The ACI Editor lets you create and modify ACIs using a visual interface or a manual editor. Depending upon your needs, you can edit visually, manually, or using both methods.

Whenever you want to work with an object's ACIs, you must use the ACI Manager. If you want to create an ACI for an object, you must also use the ACI Editor.

Each Sun ONE server may have its own uses for the ACI Editor and may have unique ACI extensions. For detailed information about a particular server's ACI options, see the documentation for that server.

What's in an ACI

Any directory entry can include one or more ACIs. Since Sun ONE servers store configuration settings, task entries, and other data as directory entries, you can apply ACIs to this information. These ACIs consist of three sections: a target, permissions, and bind rules.

Target

A target is an object, attribute, or group of objects and attributes to which you're controlling access.

Permissions

Permissions specify the rights that you are granting or denying. The permissions `Read`, `write`, and `execute` are examples of permissions that are typically specified in ACIs.

Bind Rules

Bind rules specify the circumstances under which access is allowed or denied. Bind rules may include any of the following:

- The user or group granted or denied access permission
- Host computers from which users are allowed or denied access
- An interval of time during which a user or group is allowed or denied access
- The type of permissions to grant or deny to a user or group

ACIs are stored as attributes of the target Directory Server entry. The following example illustrates the use of two ACIs in the same directory entry. The first ACI grants unrestricted access to the user directory to all members of the Directory Administrators group. The second ACI denies access to the user directory to the Directory Administrators group from 1:00 a.m. to 3:00 a.m. (0100 to 0300) on Sunday, Tuesday, and Friday. The more restrictive ACI takes control during the times specified by it. Thus, the end result is that members of the Directory Administrator's group can access the user directory at any time except between 1:00 a.m. and 3:00 a.m. on Sunday, Tuesday, and Friday.

```
dn: dc=example,dc=com
objectClass: top
objectClass: organization
ACI: (target='ldap:///dc=example,dc=com')(targetattr=*)
  (version 3.0; acl 'acl 1'; allow (all)
  groupdn = 'ldap:///cn=Directory Administrators,
  dc=example,dc=com';)
ACI: (target='ldap:///dc=example,dc=com')(targetattr=*)
  (version 3.0; acl 'acl 2'; deny (all)
  groupdn = 'ldap:///cn=Directory Administrators, dc=example,dc=com'
  and dayofweek = 'Sun, Tues, Fri' and
  (timeofday >= '0100' and timeofday <= '0300');)
```

Using the ACI Manager and ACI Editor

When you apply ACIs to tasks, user interface elements, or other directory entries, you use the ACI Manager. When setting access permissions for anything other than servers in the Sun ONE Server Console navigation tree (for instance, for tasks or user interface elements), you use the ACI Editor to create new ACIs and to modify existing ones.

While each Sun ONE server has a unique set of items that you can apply ACIs to, the ACI Manager and Editor are shared by all Sun ONE Server Console-based products. For information on a specific server's implementation of ACIs, see that server's documentation.

To Specify What You Want an ACI to Apply To

1. Select an object that you want to apply ACIs to.

    ❍ To select a task or directory entry click its name.

      Select a task name in an individual server management window. Select a directory entry in the Directory tab of the Sun ONE Directory Server management window.

- To select a user interface (UI) element, choose Preferences from the Edit menu, and then click the UI Permissions tab. On the tab, select an onscreen element from the list.

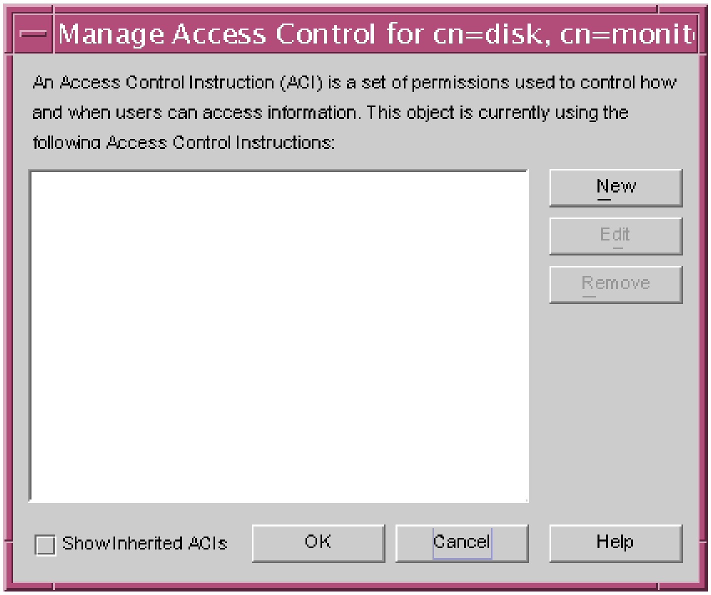
2. Open the ACI Manager.

   - To open the ACI Manager from a server management window, right-click and choose Set Access Permissions.

   - To open the ACI Manager from the UI Permissions panel of the Preferences dialog box, click the Permissions button.

   - In some servers, you can also open the ACI Manager by choosing Set Access Permissions from the Edit or Object menu.

Figure 9-3 Default ACI Manager



Manage Access Control for cn=disk, cn=monit

An Access Control Instruction (ACI) is a set of permissions used to control how and when users can access information. This object is currently using the following Access Control Instructions:

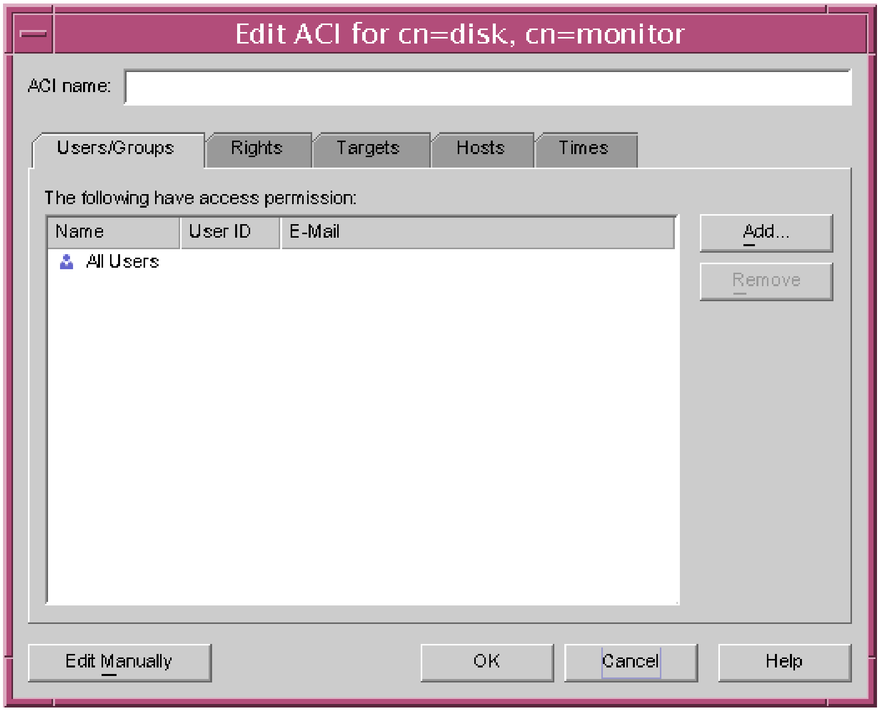
□ Show Inherited ACIs

To Create a New ACI With the Visual ACI Editor

1. In the ACI Manager click New.

   The ACI Editor appears.

Figure 9-4 Visual ACI Editor



2. Enter a name for this ACI in the ACI Name field.

3. On the Users/Groups tab, click Add.

4. Identify the users, groups, or administrators to which you want to grant access.

   - First, search for users, groups, or administrators to grant access to:

     Search for. In this field, enter the name of the user, group, or administrator that you want to add. If you do not know the full name, you can enter any part of it. To find all entries, search for *.

     Search area. Select a set of entries in which you want to search. You can choose Users and Groups, Administrators, or Special Rights.

     Search. Click this button to perform your search.

     The center frame of the Add Users and Groups dialog box displays the results of your search. This is called the results list. The bottom frame shows the users that you've granted access to. This is called the access list.

- Then, grant access:

  Click a user, group, or administrator in the results list to select it. You can select multiple entries by pressing Control and clicking the desired users and groups.

  Add. Click this button to add a selected user from the results list to the access list.

  Remove. Click this button to remove a user from the access list.

If you want to add more users or groups to the access list, you can perform additional searches.

5. Click OK.

6. On the Rights tab, specify which actions are permitted as part of this ACI. Select a single action to permit it, or click one of the following buttons:

   Check All. Click to select all rights.

   Check None. Click to deselect all rights.

   If you are creating an ACI for a user interface element, and you want to hide the element from the selected users, groups, and hosts, click Check None.

   The rights you select here apply to the users, groups, and administrators that you selected in step 4 as well as the targets, hosts, and times that you specify in steps 7-10.

7. On the Targets tab, specify the directory entry to which this ACI should apply.

   Target directory entry. In this field, enter the DN for the entry to which you want this ACI to apply. By default, the target directory entry is the currently selected object. This is the task or other resource that you selected before you opened the ACI Manager.

   This Entry. Click this button to reset the Target Directory Entry to the DN for the currently selected object.

   Browse. Click this button to locate a directory entry. This opens a directory tree. Choose the entry you want this ACI to apply to and then click OK.

   Filter for sub-entries. In this field, enter an LDAP filter to apply to any entries below the Target Directory Entry.

   An LDAP filter is useful if you want this ACI to apply to multiple entries within a branch of the directory. By default, this field is blank indicating that this ACI applies *only* to the currently selected object.

   These attributes are affected for all entries. In this list, select the attributes to which you want this ACI to apply. Users listed in this ACI can only access selected attributes.

   Check All. Click this button to select all listed attributes.

   Check None. Click this button to deselect all listed attributes. If no attributes are selected, this ACI applies to the Target Directory Entry.

8. On the Hosts tab, click Add.

9. Enter the host name or IP address that you want to grant access to, then click OK. You can use the * wildcard when specifying hosts.

10. On the Times tab, select the times during which you want to grant access to the desired users, groups, and hosts.

    Click a square to select or deselect it. If a square is blue, access is allowed at that time. If a square is white, access is not allowed at that time.

11. Click OK to save this ACI.

    If you selected a task or directory entry, the ACI is automatically applied to it. If you selected a user interface element, you must restart Sun ONE Server Console for the ACI to take effect.

To Create a New ACI With the Manual ACI Editor

1. In the ACI Manager click New.

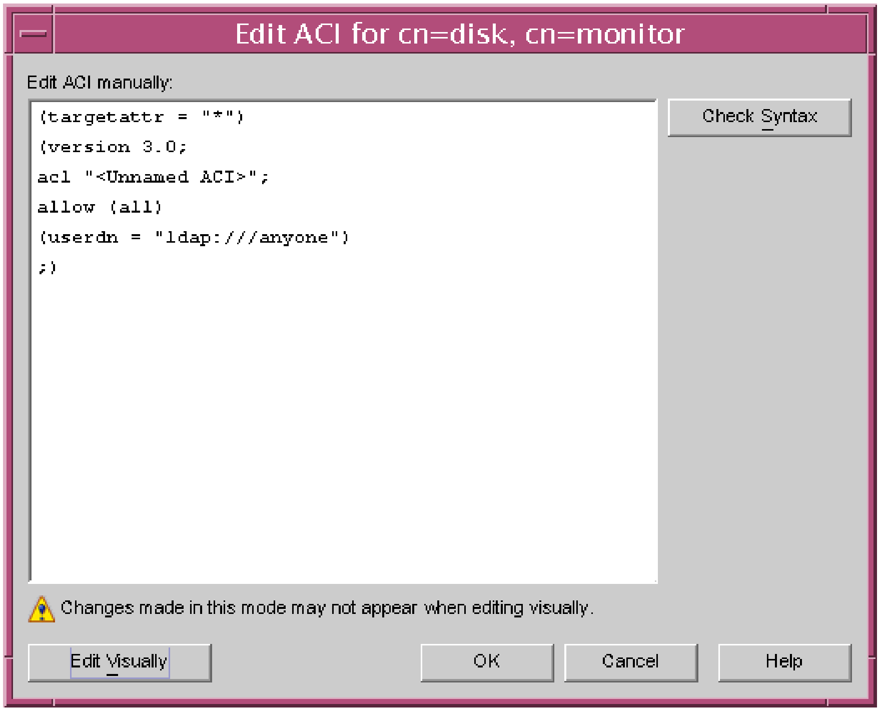
   The ACI Editor appears.

2. Enter a name for this ACI in the ACI Name field.

3. Click Edit Manually.

   The ACI Editor switches into manual mode.

Figure 9-5 Manual ACI Editor



```
(targetattr = "*")
(version 3.0;
acl "<Unnamed ACI>";
allow (all)
(userdn = "ldap:///anyone")
;)
```

Changes made in this mode may not appear when editing visually.

4. Enter your ACI.

5. (Optional) Click Check Syntax to verify that your ACI is in the correct format.

NOTE If you decide you'd prefer to edit your ACI using the visual ACI Editor, you can do so by clicking Edit Visually. You may not be able to edit all ACI properties visually. To return to the manual ACI Editor, click Edit Manually. What you created visually appears in the manual editing window and you can add to it.

6. When you have finished creating your ACI, click OK.

   If you selected a task or directory entry (in "To Specify What You Want an ACI to Apply To," on page 134), the ACI is automatically applied to it. If you selected a user interface element, you must restart Sun ONE Server Console for the ACI to take effect.

To Edit an Existing ACI With the ACI Editor

1. In the ACI Manager, select the ACI that you want to modify. Click Edit.

   The ACI Editor appears.

2. Make the desired changes.

   Use the visual ACI Editor or the manual ACI Editor just as you did to add an ACI. For more information, see the procedures for adding an ACI above.

3. When you are finished, click OK.

   If the ACI was for a task or directory entry, the ACI is automatically applied to the task or entry. If the ACI was for a user interface element, you must restart Sun ONE Server Console for the ACI to take effect.

To Remove an ACI

1. In the ACI Manager, select the ACI that you want to remove.

2. Click Remove.

3. Click OK to remove the ACI.

   If the ACI was for a task or directory entry, the ACI is automatically removed from the task or entry. If the ACI was for a user interface element, you must restart Sun ONE Server Console for the removal to take effect.

Chapter 10

Using SSL and TLS with Sun ONE Servers

This chapter describes how to set up support for the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols in Sun ONE servers. Before reading this chapter, you should be familiar with the concepts described in Appendix A, "Introduction to Public-Key Cryptography."

This chapter contains the following sections:

- The SSL and TLS Protocols
- Preparing to Use SSL and TLS Encryption
- Obtaining and Installing a Server Certificate
- Activating SSL
- Managing Server Certificates
- Using Client Authentication

The SSL and TLS Protocols

The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols are sets of rules governing server authentication, client authentication, and encrypted communication between servers and clients. SSL and TLS are widely used on the Internet, especially for interactions involving the exchange of confidential information such as credit card numbers.

At a minimum, SSL and TLS require a server certificate. As part of the initial “handshake” process, the server authenticates its identity by presenting this server certificate to the client. Using public-key encryption and digital signatures, the client confirms that the server is, in fact, the server it claims to be. If desired, the server can also request that the client authenticate its identity by presenting a client certificate.

If authentication is successful, the client and server use techniques of symmetric-key encryption to encode all the information they exchange for the remainder of the session. Symmetric-key encryption also allows the client and server to detect if any tampering has occurred during the transmission of data.

SSL and TLS Ciphers

The SSL and TLS protocols support a variety of different cryptographic algorithms for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. These algorithms are called *ciphers* and are often implemented in sets called *cipher suites*. Clients and servers may support different cipher suites depending on factors such as the version of SSL or TLS they use, and company policies regarding acceptable encryption strength. Among their other functions, the SSL and TLS protocols determine how servers and clients negotiate which cipher suites they use to communicate.

Each new version of SSL and TLS maintains backward compatibility with earlier versions. As a result, the SSL 2.0, SSL 3.0, and TLS protocols have several cipher suites in common. This allows a newer client or server to communicate securely with an older client or server. To control the level of encryption used during communication, administrators can enable or disable cipher suites on both clients and servers. When a particular client and server exchange information during the SSL or TLS handshake, they identify the strongest enabled cipher suites they have in common and use those for the session.

Choosing SSL and TLS Ciphers

Decisions about which cipher suites an organization enables are often based on both the sensitivity of the data involved and the speed of the cipher. A 40-bit cipher is relatively easy to break, but very fast. A 128-bit cipher is difficult to break, but slower than other ciphers.

Some organizations may want to disable less secure ciphers to prevent insufficiently encrypted SSL connections. To serve the greatest number of users, it is a good idea for administrators to enable as broad a range of SSL cipher suites as possible. That way, when clients or servers are dealing with each other, they can negotiate the use of the strongest ciphers available.

Since 40-bit ciphers can be broken relatively quickly, administrators whose user communities can use stronger ciphers should disable all 40-bit ciphers if they are concerned about access to data by eavesdroppers.

For detailed information on determining which cipher suites to use when setting up SSL, see Appendix B, “Introduction to SSL.”

Preparing to Use SSL and TLS Encryption

All Sun ONE servers support PKCS #11 and the SSL protocol. Many Sun ONE servers also support TLS. Before you request certificates and begin to exchange information securely, you’ll need to set up SSL and TLS. If you’re using an external security device, you also need to install a PKCS #11 module.

Using External Security Devices

External security devices are Public Key Cryptography Standard (PKCS) #11 modules. PKCS defines the interface used for communication between SSL and PKCS #11 modules.

A PKCS #11 module is a device, implemented in hardware or software, that provides cryptographic services such as encryption, decryption and, in some cases, storage of keys and certificates. All Sun ONE servers include a built-in software PKCS #11 module.

Sun ONE servers can use a variety of external PKCS #11 modules provided by different manufacturers including for example Sun Crypto Accelerator Boards. Before using an external module, you must install the manufacturer’s drivers on the machine running your Sun ONE server.

Slots and Security Devices

A PKCS #11 module always has one or more slots. Slots can be implemented physically in a piece of hardware or conceptually in software. Each slot in a PKCS #11 module can contain a *security device*, the hardware or software that actually provides cryptographic services and stores certificates and keys. For example, a smart card reader contains one or more slots, each of which can contain a security device called a smart card.

An *internal security device* is made up of a key-pair and a certificate database stored in a software file on a host computer. By default, Sun ONE Administration Server provides a means to create an internal security device with its PKCS #11 module. If you do not have an external device connected to your server or client, you can use only the Sun ONE internal security device for SSL authentication.

An *external security device* is a key-pair and certificate database stored in an external device such as a Smart Card. If you have an external device connected to your server, you can use internal and external security devices for SSL authentication.

To Install an External Security Device

1. Connect your Smart Card reader or other device and install its drivers on your host machine.

   Initially, the device is available to all servers on the host. Depending on the device's capabilities, you may be able to share it across multiple servers on the host. For more information, see the documentation that came with your hardware.

2. In the Sun ONE Server Console navigation tree, select the server instance that you want to use the PKCS #11 module with, and then click Open.

3. From the server's Console menu, choose Security > Configure Security Modules, and then click Install.

4. In the Install Security Module dialog box, enter the following information:

   Enter the PKCS #11 module driver filename. Enter the full path to the driver file that came with your device. This file has the extension `DLL`, `JAR`, `so`, or `sl`.

   Enter an identifying name for this module. Enter a descriptive name that helps you identify this device.

5. Click OK, and then click Close.

To Remove an External PKCS #11 Module

1. In the Sun ONE Server Console navigation tree, select the server instance that is using the external PKCS #11 device, and then click Open.

2. From the server's Console menu, choose Security > Configure Security Modules.

3. Select your device from the list and then click Remove.

4. Click OK to confirm that you want to remove the device, and then click Close.

Obtaining and Installing a Server Certificate

When requesting and installing certificates, you use two wizards. You use the Certificate Request Wizard to request a new server certificate or to renew a certificate that you’re already using. You use the Certificate Installation Wizard to install a certificate that you’ve received from a *Certificate Authority* (CA). The first time you use the Certificate Request Wizard, it also creates and installs a *key and certificate database* for you.

This section takes you through the steps of requesting and installing a certificate.

SSL Certificates

Sun ONE Server Console can install three types of certificates: server certificates, server certificate chains, and trusted CA certificates.

A *server certificate* is a single certificate associated only with your server. It identifies your server to clients. You must request this type of certificate from a CA. To obtain and install a Server Certificate, generate a request and send it to the CA. Then install the certificate.

For information on installing a server certificate, see “Generating a Server Certificate Request,” on page 146 and “Installing the Certificate,” on page 149.

A *server certificate chain* is a collection of certificates automatically generated for you by your company’s internal certificate server or a known CA. The certificates in a chain trace back to the original CA, providing proof of identity. This proof is required each time you obtain or install a new server certificate.

A *trusted CA certificate* is a single certificate automatically generated for you by your company’s internal certificate server or a known CA. A trusted CA certificate is used to authenticate clients.

To obtain a trusted CA certificate, first go to the internal certificate server or CA’s web site. Copy the necessary certificate information and save it to a file. Then use the Certificate Installation Wizard to install the certificate. For more information, see “Installing the Certificate,” on page 149.

You can install any number of SSL certificates on a server. When setting up SSL for an instance of Directory Server, you need to install at least a server certificate and a trusted CA certificate.

Preparing to Set Up SSL and TLS

You need to set up SSL and TLS differently depending on whether you are using an internal security device, an external hardware device, or both. This section tells you how to do this.

Setting up SSL or TLS With an Internal Security Device

To set up SSL or TLS with an internal security device, you must request and install a certificate. To request a certificate, run the Certificate Request Wizard. To install the certificate, run the Certificate Installation Wizard. When prompted, specify that you want to install the certificate on the internal security device.

Setting up SSL or TLS With an External Security Device

To set up SSL with an external security device first install the PKCS #11 module provided by the external device manufacturer. Then run the Certificate Request Wizard, specifying the external security device when prompted. For more information, see "To Install an External Security Device," on page 144.

Setting Up SSL With Internal and External Security Devices

Some servers and clients in your enterprise may use only internal security devices, while others may use both internal and external security devices. If your server needs to communicate with products running both internal and external security devices, run the Certificate Request Wizard *two times*. During the first use, when prompted, specify the internal security device. During the second use, when prompted, specify the external security device.

Generating a Server Certificate Request

You can use Sun ONE Server Console to generate a certificate request which you can then submit to a CA.

At this time, you must provide a password for the security device. After installing the server certificate, this password must be provided at server startup.

To Generate a Certificate Request

1. In the Sun ONE Server Console navigation tree, select the server instance with which you want to use SSL encryption.

2. Double click the server instance or click Open to open the management window for the server instance.

3. From the Console menu, choose Security > Manage Certificates.

   Alternatively, you can click the Manage Certificates task.

4. Enter the password for the security device that holds this certificate.

   If you are installing the certificate on the internal (software) security device, enter the password for the key and certificate database. If you are installing a certificate on an external (hardware) security device, enter the password for the device.

5. Click the Server Certs tab.

6. Click Request to open the Certificate Request Wizard.

7. Choose "Request Certificate Manually," and then click Next.

8. Enter the requested information:

   Server Name. (Optional) Enter the fully qualified host name of the machine for which you're requesting a certificate.

   Organization. (Optional) Enter your organization's name.

   Organizational Unit. (Optional) Enter your division, department, or other organizational unit.

   City/locality. (Optional) Enter the city or locality in which your organizational unit is located.

   State/province. (Optional) Enter the state or province in which your organizational unit is located.

   Country/region. (Optional) Select the state or province in which your organizational unit is located, from the drop down menu.

   You can toggle between two views of the request form using the following buttons:

   Show DN. Click to show the requestor information in distinguished name (DN) format. The Show DN button is visible only when you are entering information in fields.

   Show Fields. Click to show the requestor information in fields. The Show Fields button is visible only when you are entering information in DN format.

9. Click Next.

10. Enter the password for the security device that stores this certificate.

    If you are requesting the certificate for an internal (software) security device, this is the password for the key and certificate database. If you are requesting the certificate for an external (hardware) module, this is the password for your SmartCard or other security device.

NOTE You must generate a different request for each device.

11. Click Next.
12. Select one of the following:

    Copy to Clipboard. Click to copy your certificate request to the clipboard.

    Save to File. Click to save your request as a text file. You are prompted to choose a name and location for the file.

    The certificate request you have copied to the clipboard or saved as a text file is required to mail to a Certificate Authority issuing the new certificate. See "To Send a Server Certificate Request as Email," on page 148.

13. Click Done to close the Certificate Request Wizard.

Sending a Server Certificate Request

Once you have generated a server certificate request, you send it to a CA for processing. Many CAs allow you to submit certificate requests through their web sites. Others may require you to send them an email message containing your request.

To Send a Server Certificate Request as Email

1. Use your email program to create a new email message.
2. Paste your certificate request into the message.

   If you copied the certificate request to the clipboard, paste it into the body of the message.

   If you saved your certificate request to a file, open it in a text editor. Copy and paste the request into the body of the message.

3. Enter a subject and recipient for your request. The type of subject and recipient varies depending on which CA you are using. For more information, see your CA's web site.

4. Send the email message to the CA.

Once you've submitted your request, you must wait for the CA to respond with your certificate. Turnaround time is highly variable and depends on the CA. If your company has an internal CA, it may take only a day or two to receive your certificate. If you are using an external CA, it could take as long as several weeks for that CA to respond to your request.

Installing the Certificate

Depending on the CA, you may receive your certificate in an email message or you may have to retrieve it from the CA's web site. Once you have the certificate, you can back it up and install it.

To Back Up a Certificate

- Save, in a text file, the certificate data you received from the CA.

  If you ever lose the certificate data, you can reinstall the certificate using this backup file.

To Install a Server Certificate

When you install the first server certificate, you are prompted for a password for the certificate database.

1. In the Sun ONE Server Console navigation tree, select the server instance on which you want to install the certificate.

2. Click Open to open the management window for the server instance.

3. On the Tasks tab, click the Manage Certificates task button.

   Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.

4. Click the Server Certs tab.

5. Specify where to store this certificate.

   ❍ If you want to store this certificate on the internal security device, select internal (software) from the Security Device drop-down list.

- If you want to store this certificate on an external hardware device, select the device from the Security Device drop-down list.

6. Click Install.
7. Enter the certificate's location or enter its text.

   In this local file. If your certificate is stored in a text file on your system, enter the full path to the file.

   In the following encoded text block. If you copied your certificate to the clipboard, paste the certificate's text into the text field by clicking the Paste from Clipboard button.
8. Click Next.

   If the certificate information you entered above is valid, you see a page containing the details of your certificate.
9. Verify that the certificate information is correct, and then click Next.
10. Enter a name for the certificate, and then click Next.
11. Enter the password for the security device that holds this certificate.

    If you are installing the certificate on the internal (software) security device, enter the password for the key and certificate database. If you are installing a certificate on an external (hardware) security device, enter the password for the device.
12. Click Done.

To Install a CA Certificate or Server Certificate Chain

1. Obtain the CA certificate or Server Certificate Chain from your CA.
2. In the Sun ONE Server Console navigation tree, select the server instance on which you want to install the CA certificate.
3. Click Open to open the management window for the server instance.
4. On the Tasks tab, click the Manage Certificates task button.

   Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.
5. Select the CA Certs tab, and then click Install.

6. Enter the certificate’s location or enter its text:

   In this local file. If the certificate is stored in a text file on your system, enter the full path to the file.

   In the following encoded text block. If you copied the certificate to the clipboard, paste the certificate’s text into the text field by clicking the Paste from Clipboard button.

7. Click Next.

   If the certificate information you entered in step 6 is valid, you see a page containing the details of the certificate.

8. Verify that the certificate information is correct, and then click Next.

9. Enter a name for the certificate, and then click Next.

10. Select the trust options for this certificate:

    Accepting Connections from Clients. Check this box if you want to trust client certificates issued by this CA.

    Making Connections to Other Servers. Check this box if you want to trust server certificates issued by this CA.

11. Click Done.

Backing Up and Restoring Your Certificate Database

Whenever you install a certificate, you should back up your certificate database. If your database ever becomes corrupted, you can restore your certificate information from this backup.

To Back Up Your Certificate Database

1. Open your server root folder.

2. Copy all files in the `alias` folder to another location (preferably on a different disk).

   This folder includes your certificates as well as the private key for your trust database.

To Restore Your Certificate Database From a Backup

- Copy your backup files to the `alias` subfolder of your server root folder.

CAUTION If you restore your certificate database from a backup, any certificates that you installed after making the backup are lost. Before restoring your certificate database, make sure that you have copies of all your certificates in case you need to reinstall them.

Activating SSL

Once you've obtained and installed a server certificate, use Sun ONE Server Console to activate SSL on your Sun ONE server. The following procedure uses Sun ONE Administration Server as its example. Activating SSL on other Sun ONE servers is done the same way, although in some cases the interface is slightly different. For more information on how to activate SSL on another server product, see that server's documentation.

To Activate SSL on a Sun ONE Server

1. In the Sun ONE Server Console navigation tree, select the server instance with which you want to use SSL encryption.
2. Click Open to open the management window for the server instance.
3. Click the Configuration tab.
4. Click the Encryption tab.

5. Enter information as appropriate:

   Enable SSL for this server. Select this option if you want to secure this server with Secure Sockets Layer (SSL) encryption. All other SSL encryption options listed here become available to you only when you enable SSL by checking this box.

   Use this cipher family. When you enable SSL encryption, the cipher families available to you are listed here. Sun ONE Server Console currently supports RSA cipher families. Select the cipher families you want to use.

   Security Device. Choose internal (software) if the key is stored in the local key database. All other choices on this list are available only if you are using an external module.

   Certificate. Choose a server certificate to use with this server.

   Settings. Click this button to modify cipher (encryption algorithm) settings for the certificate you selected above.

   Disable Client Authentication. Select this option if you do not want this server instance to perform client authentication.

   Require Client Authentication. Select this option if you want this server instance to require client authentication during the SSL handshake. For background information, refer to “Using Client Authentication,” on page 160.

   If you select this option, each Sun ONE Server Console administrator is prompted for a certificate when logging in. This ensures system security because all administrators must present acceptable certificates before they can perform management tasks. Even if an intruder obtains a user name and password, the intruder must present a valid certificate (one issued by a trusted CA) to gain access to your enterprise.

   For more information on setting trust options for CA certificates, see “To Change the CA Trust Options,” on page 156.

NOTE Specific servers include additional options here.

6. Click the Network tab.
7. Set the port number for the Secure Port.
8. Click Save.

9. Exit Sun ONE Server Console and restart the server you have SSL-enabled from the command line.

   You can now start Sun ONE Server Console again and log in to work with the server through Sun ONE Server Console.

Managing Server Certificates

Periodically, you may need to update information for your installed SSL certificates. From Sun ONE Server Console, you can renew a server certificate as well as view and edit settings for all certificates installed on a server.

Renewing a Certificate

Like credit cards or any other form of identification, all certificates have validity periods. You can check any certificate's expiration date from within Sun ONE Server Console. When a server certificate is nearing its expiration date, you can use Sun ONE Server Console to generate a renewal request.

To Check a Certificate Expiration Date

1. In the Sun ONE Server Console navigation tree, select the server instance that is using the certificate whose expiration date you want to check.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.

   Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.
4. Depending on which type of certificate you are checking, click the Server Certs or CA Certs tab.
5. Locate the certificate you are checking.

   The certificate's validity period ends on the date shown in the Expiration Date column.

To Generate a Certificate Renewal Request

1. In the Sun ONE Server Console navigation tree, select the server instance that is using the certificate you want to renew.

2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.

   Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.
4. Click the Server Certs tab.
5. From the list of available certificates, select the one you want to renew, and then click the Renew button.
6. Select “Request Certificate Manually,” and then click Next.
7. Enter the requested information:

   Server Name. (Optional) Enter the fully qualified host name of the machine for which you’re requesting a certificate.

   Organization. (Optional) Enter your organization’s name.

   Organizational Unit. (Optional) Enter your division, department, or other organizational unit.

   City/locality. (Optional) Enter the city or locality in which your organizational unit is located.

   State/province. (Optional) Enter the state or province in which your organizational unit is located.

   Country/region. (Optional) Enter the state or province in which your organizational unit is located.

   You can toggle between two views of the request form using the following buttons:

   Show DN. Click to show the requestor information in distinguished name (DN) format. This button is visible only when you are entering information in fields.

   Show Fields. Click to show the requestor information in fields. This button is visible only when you are entering information in DN format.
8. Click Next.
9. Enter the password for the security device that stores this certificate.

   If you are using the internal (software) security device, this is the password for the key and certificate database. If you are using an external (hardware) module, this is the password for your SmartCard or other security device.

10. Click Next.

11. Copy or save the request in one of the following ways:

    Copy to Clipboard. Click to copy your certificate request to the clipboard.

    Save to File. Click to save your request as a text file. You are prompted to choose a name and location for the file.

    The certificate request you have copied to the clipboard or saved as a text file is required to email to a Certificate Authority issuing the new certificate. See "To Send a Server Certificate Request as Email" on page 148.

12. Click Done to close the Certificate Request Wizard.

You can now send your certificate renewal request to your CA. For more information, see "To Send a Server Certificate Request as Email," on page 148.

Changing the CA Trust Options

At times, you may need to reject a generally trusted CA. For example, if you are notified that a CA is experiencing technical difficulties that prevent certificate authentication, you can temporarily reject the CA's certificate. When you are informed that the problem has been resolved, you can begin trusting the certificate again.

To Change the CA Trust Options

1. In the Sun ONE Server Console navigation tree, select the server instance on which you want to change a CA trust option.

2. Click Open to open the management window for the server instance.

3. On the Tasks tab, click the Manage Certificates task button.

   Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.

4. Click the CA Certs tab and then, from the list of available CA certificates, select the CA certificate for which you want to change the trust options.

5. Click the Edit Trust button.

6. Set the following CA trust options:

   Accepting connections from clients (Client Authentication). Uncheck this box if you want to reject client certificates issued by this CA.

   Making connections to other servers (Server Authentication). Uncheck this box if you want to reject server certificates issued by this CA.

7. Click OK.

Changing Security Device Passwords

You should periodically change the passwords for your security devices.

To Change a Security Device Password

1. In the Sun ONE Server Console navigation tree, select the server instance that is using the security device for which you want to change the password.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.

   Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.
4. Choose a security device from the drop-down list.
5. Click Password.
6. In the Change Security Device Password dialog box, enter password information:

   Old password. Enter the password currently used with this device.

   New Password. Enter a new password.

   New Password (again). Enter the password again to confirm it.
7. Click OK.

Managing Revoke Certificate Lists

Certificate revocation lists (CRLs) and compromised key lists (CKLs) allow CAs to specify certificates and keys that client or server users should no longer trust.

If data in a certificate changes, a CA can revoke the certificate and list it in a CRL—for example, when a user changes offices or leaves an organization before his or her certificate expires. If a key is tampered with or otherwise compromised, a CA can list it in a CKL.

CRLs and CKLs are produced and periodically updated by a CA.

To Obtain a CRL or CKL From a CA

1. Use a browser to go to the CA's web site. Contact your CA administrator for the exact URL.

2. Follow the CA's instructions for downloading the CRL or CKL to a local directory.

Once you've saved the CRL or CKL file to a local directory, you can add its contents to the database of lists of revoked certificates and compromised keys. Once you do this, your server no longer trusts the certificates or keys that are specified in the CRL or CKL file.

To View, Add, or Delete a CRL or CKL

1. In the Sun ONE Server Console navigation tree, select the server instance that you want to work with.

2. Click Open to open the management window for the server instance.

3. On the Tasks tab, click the Manage Certificates task button.

   Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.

4. Choose a security device.

   If the server instance is using only the internal (software) security device, it is automatically chosen for you. If you are using an external (hardware) module, choose it from the drop-down list.

5. Select the Revoked Certs tab.

   Every CRL and CKL for the chosen device is listed along with the date it was generated and the date for the next update.

6. View, add, or delete a CRL or CKL.

   ❍ To view the contents of a CRL or CKL, select its name, and click Detail.

- To add a CRL or CKL for the selected device, click Add, and then enter the following information:

  Enter full path to CRL/CKL file. Provide the full path to the file containing the CRL or CKL.

  File contains a Certificate Revocation List (CRL). Select this option if you're adding a CRL.

  File contains a Compromised Key List (CKL). Select this option if you're adding a CKL.

- To delete a CRL or CKL from the selected device's trust database, select it, and then click Delete.

7. Click OK.

Using Client Authentication

You can configure some Sun ONE servers to require that clients present certificates when logging in. This allows a server to verify a client's authenticity and to determine if a user has access to the server. The process of presenting and verifying a client certificate is called client authentication.

This section tells you how to set up and use client authentication on your Sun ONE server. Before reading this section, check your server's documentation to verify that the server supports client authentication.

How Client Authentication Works

When a server receives a request from a client, it can ask for the client's certificate before proceeding. A client is programmed to respond by sending a client certificate to the server.

After checking that a client certificate chain ends with a trusted CA, an Sun ONE server can optionally determine which user is identified by the client certificate and then look up that user's entry in the directory. The server authenticates the user by comparing the information in the certificate with the data in the user's directory entry.

In order to locate user entries in the directory, a server must know how to interpret certificates from different CAs. You provide the server with interpretation information by editing a file called `certmap.conf`. This file provides three kinds of information for each listed CA:

- It maps the distinguished name (DN) in the certificate to a branch point in the LDAP directory.
- It specifies which DN values from the certificate (user name, email address, and so on) the server should use for the purpose of searching the directory.
- It specifies whether the server should go through an additional verification process. This process involves comparing the certificate presented by the client for authentication with the certificate stored in the user's directory entry. By comparing the certificate, the server determines whether to allow access or whether to revoke a certificate by removing it from the user's directory entry.

If more than one directory entry contains the information in the user's certificate, the server can examine all matching entries in order to determine which user is trying to authenticate. When examining a directory entry, the server compares the presented certificate with the certificate stored in the entry. If the presented certificate doesn't match any directory entries or if matching entries don't contain matching certificates, client authentication fails.

After the server finds a matching entry and certificate in the directory, it can determine the appropriate kind of authorization for the client. For example, some servers use information from a user's entry to determine group membership, which in turn can be used during evaluation of ACIs to determine what resources the user is authorized to access.

You can also configure client authentication between an instance of Administration Server and another Sun ONE server. For more information see "Using Client Authentication Between Servers," on page 167."

Preparing to Use Client Authentication

In order to accept certificates for client authentication, you must fulfill the following requirements:

- The server must have SSL turned on. For more information, see "Activating SSL," on page 152.
- The instance of Administration Server must trust the CA who issued the certificate to the client. For more information, see "Changing the CA Trust Options," on page 156.
- If you are going to search the directory for information contained in certificates, you must map specific CAs to branches of the user directory. To do this, you must edit a file called `certmap.conf`. The rest of this section describes this file and tells you how to edit it.

The certmap.conf File

When a server performs client authentication, it interprets a certificate, extracts user information, and then searches the directory for that information. In order to process certificates from different CAs, the server uses a file called `certmap.conf`. This file contains instructions on how to interpret different certificates and how to search the directory for the information that those certificates contain.

The `certmap.conf` file is stored in the *ServerRoot*`/shared/config` folder. The file contains a default mapping as well as mappings for specific CAs.

The default mapping specifies what the server should do if a client certificate was issued by a CA that isn't listed in `certmap.conf`. The mappings for specific CAs specify what the server should do for client certificates issued by those CAs. All mappings define the following:

- Where in the directory the server should begin its search
- What certificate attributes the server should use as search criteria
- Whether the server should verify the certificate with a certificate that is stored in the directory

Mappings have the following syntax:

```
certmap name issuerDN
name:property [value]
name:property [value]
...
```

The first line of a mapping specifies the mapping's name as well as the DN for the issuer of the client certificate. You can name a mapping whatever you want, but the `issuerDN` must exactly match the issuer DN of the CA that issued the client certificate. For example, the following two `issuerDN` lines differ only in the number of spaces they contain, but the server would treat these two entries as different:

```
certmap name ou=Sun ONE CA,o=Sun ONE,c=US
certmap name ou=Sun ONE CA, o=Sun ONE, c=US
```

The second and subsequent lines of a mapping identify the rules that the server should use when searching the directory for information extracted from a certificate. These rules are specified through the use of one or more of the following properties: `DNComps`, `FilterComps`, `VerifyCert`, `CmapLdapAttr`, `Library`, and `InitFn`. These properties are explained next.

DNComps

DNComps is a comma-separated list of relative distinguished name (RDN) keywords used to determine where in the user directory the server should start searching for entries that match the information for the owner of the client certificate. The server gathers values for these keywords from the client certificate and uses the values to form a DN, which determines where the server starts its search in the directory.

For example, if you set DNComps to use the o and c RDN keywords, the server starts the search from the o=org, c=country entry in the directory, where org and country are replaced with values from the DN in the certificate.

- If there isn't a DNComps entry in the mapping, the server uses either the CmapLdapAttr setting or the entire subject DN in the client certificate to determine where to start searching.
- If the DNComps entry is present but has no value, the server searches the entire directory tree for entries matching the filter specified by FilterComps.

The following RDN keywords are supported for DNComps: cn, ou, o, c, l, st, e, and mail. You can list the keywords in lower case or upper case. You can use e or mail, but not both.

FilterComps

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

For example, if FilterComps is set to use the e and uid attribute keywords (FilterComps=e,uid), the server searches the directory for an entry whose values for e and uid match the user's information gathered from the client certificate. Email addresses and user IDs are good filters because they are usually unique entries in the directory.

The filter needs to be specific enough to match one and only one entry in the directory. The following RDN keywords are supported for FilterComps: cn, ou, o, c, l, st, e, and mail. You can list the keywords in lowercase or uppercase letters. You can use e or mail, but not both.

VerifyCert

`VerifyCert` tells the server whether it should compare the client's certificate with the certificate found in the user's directory entry. It takes one of two values: `on` or `off`. Setting the value to `on` ensures that the server does not authenticate the client unless the certificate presented exactly matches the certificate stored in the directory. Setting the value to `off` disables the verification process.

CmapLdapAttr

`CmapLdapAttr` is the name of the attribute in the directory that contains subject DNs from all certificates belonging to the user. Because this attribute isn't a standard LDAP attribute, you have to extend the LDAP schema to include it.

If the `CmapLdapAttr` property exists in a `certmap.conf` mapping, the server searches the entire directory for an entry that contains the subject's full DN. The search criteria are the attribute named by `CmapLdapAttr` and the subject's full DN as listed in the certificate. If the search doesn't yield any entries, the server retries the search using the `DNComps` and `FilterComps` mappings. The search takes place more quickly if the attribute specified by `CmapLdapAttr` is indexed.

Using `CmapLdapAttr` to match a certificate to a directory entry is useful when it's difficult to match entries using `DNComps` and `FilterComps`.

Library

`Library` is the pathname to a shared library or DLL. You need to use this property only if you want to extend or replace the standard functions that map information in `certmap.conf` to entries in the directory. This property is typically not necessary unless you have very specialized mapping requirements.

InitFn

`InitFn` is the name of an `init` function from a custom library. You need to use this property only if you want to extend or replace the functions that map information in `certmap.conf` to entries in the directory. This property is typically not necessary unless you have very specialized mapping requirements.

Editing the certmap.conf File

This section tells you how to edit the `certmap.conf` file.

To Edit the certmap.conf File

1. In a text editor, open *ServerRoot*`/shared/config/certmap.conf`.

2. If necessary, make changes to the default mapping.

   For example, you may want to change the value for `DNComps` or `FilterComps`. If you want to comment out a line, insert a # before it.

3. If desired, create a mapping for a specific CA.

   The mapping should take this form: `certmap` *mappingName issuerDN*.

   For example, to create a mapping named "Example CA" which has the issuer DN `ou=Example CA, o=Example, c=US`, you would enter the following:

   ```
   certmap Example CA   ou=Example CA, o=Example, c=US
   ```

4. Add property settings for a specific CA's mapping.

   If you are using the `library` and `InitFn` properties, you must specify them before adding any additional properties.

   When adding a property, use this form:

   mappingName:*propertyName value*

   For example, you could add a `DNComps` value of `o, c` for Example CA by entering the following line:

   ```
   Example CA:DNComps   o, c
   ```

   If you are using the `Library` and `InitFn` properties, a complete mapping might look like this:

   ```
   certmap Example CA      ou=Example CA, o=Example, c=US
   Example CA:Library      /usr/ds/v5.2/servers/userdb/plugin.so
   Example CA:InitFn       plugin_init_dn
   Example CA:DNComps      o, c
   Example CA:FilterComps  e, uid
   Example CA:VerifyCert   on
   Example CA:CmapLdapAttr certSubjectDN
   ```

5. Save the `certmap.conf` file.

Example certmap.conf Mappings

The following examples illustrate three different ways you can use the `certmap.conf` file.

Example of a Default Mapping

Here are the contents of a simple `certmap.conf` file that contains only the default mapping:

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

Using this example, the server starts its search at the directory branch point containing the entry ou=*organizationalUnit*, o=*organization*, c=*country*, where the italics represent values from the subject's DN in the client certificate.

The server then uses the values for `e` (email address) and `uid` (user ID) from the certificate to search for a match in the directory before authenticating the user. When it finds a matching entry, the server verifies the certificate by comparing the certificate the client sent to the certificate stored in the directory.

Example of an Additional Mapping

Here are the contents of a sample `certmap.conf` file that defines a default mapping as well as a mapping for MyCA:

```
certmap default     default
default:DNComps
default:FilterComps e, uid
certmap MyCA        ou=MySpecialTrust,o=MyOrg,c=US
MyCA:DNComps        ou,o,c
MyCA:FilterComps    e
MyCA:verifycert     on
```

When the server gets a certificate from a CA other than MyCA, the server uses the default mapping, which starts at the top of the directory tree and searches for an entry matching the client's email address (`e`) and user ID (`uid`). If the certificate is from MyCA, the server starts its search at the directory branch containing the organizational unit specified in the subject DN and searches for email addresses (`e`) that match the one specified in the certificate. If the certificate is from MyCA, the server verifies the certificate. If the certificate is from another CA, the server does not verify it.

Example of a Mapping With an Attribute Search

This example uses the `CmapLdapAttr` property to search the directory for an attribute called `certSubjectDN` whose value exactly matches the entire subject DN in the client certificate:

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

If the subject DN in the client certificate is `uid=Henry Jones Junior, o=Example Inc, c=US`, then the server searches for entries that have `certSubjectDN=uid=Henry Jones Junior, o=Example Inc, c=US`.

If one or more matching entries are found, the server proceeds to verify the entries. If no matching entries are found, the server uses `DNComps` and `FilterComps` to search for matching entries. For the client certificate described above, the server would search for `uid=Henry Jones Junior` in all entries under `o=Example Inc, c=US`.

Using Client Authentication Between Servers

If both servers support client authentication, you can use client authentication when establishing a connection from one Sun ONE server to another. Typically, you use client authentication to authenticate an instance of Administration Server to another Sun ONE server instance. In these cases, the instance of Administration Server acts as the client.

The following procedure tells you how to set up client authentication between an Sun ONE server and an instance of Administration Server.

To Set Up Client Authentication Between Servers

1. Install certificates on an instance of Administration Server and the Sun ONE server instance that performs the authentication.

   For more information, see “To Install a Server Certificate,” on page 149.

2. If necessary, install CA certificates and specify that they should be trusted.

   The instance of Administration Server needs to trust the CA that issued the certificate in use by the Sun ONE server instance. The Sun ONE server instance needs to trust the CA that issued the certificate in use by the instance of Administration Server.

   For more information, see “To Install a CA Certificate or Server Certificate Chain,” on page 150.

3. On the Sun ONE server instance that performs the authentication, enable SSL and Client Authentication, and then restart the server.

   Typically, you enable SSL and Client Authentication by changing the encryption settings on the server's Configuration tab. For more information, see your server's documentation.

4. In a text editor, open *ServerRoot*`/admin-serv/config/adm.conf`.

5. Change the value for `ldapPort` to the secure port in use by the Sun ONE server instance.

6. Restart the instance of Administration Server.

   For more information, see "Restarting Administration Server," on page 91.

The Sun ONE server instance now uses client authentication when communicating with the instance of Administration Server.

Client Authentication for Users

You can use client authentication to verify the identity and access permission of a user, typically an administrator, to an Administration Server instance. Before enabling client authentication for users, the server must have a CA certificate chain and server certificate installed and have SSL enabled.

Instructions for obtaining and installing server certificates and CA certificate chains are found in this chapter in the section entitled "Obtaining and Installing a Server Certificate," on page 145. Instructions for enabling SSL are also found in this chapter in the section entitled "Activating SSL," on page 152.

NOTE New and existing certificates are not recognized by Administration Server unless they are stored in the Netscape Navigator 4.7x certificate database format. For initial setup of client authentication, you may store certificates in the Netscape Navigator 4.7x browser.

To Set Up Client Authentication for Users

1. Install certificates on both the instance of Administration Server and the client that participates in authentication.

   For more information, see "To Install a Server Certificate," on page 149.

2. If necessary, install CA certificates and specify that they should be trusted.

   The instance of Administration Server needs to trust the CA that issued the certificate in use by the client. The client needs to trust the CA that issued the certificate in use by the Administration Server.

   For more information, see "To Install a CA Certificate or Server Certificate Chain," on page 150.

3. On the Administration Server instance that performs the authentication, enable SSL and Client Authentication, and then restart the server.

   Typically, this is done by changing the encryption settings on the server's Configuration tab. For more information, see your server's documentation.

4. Save client certificates in the Netscape Communicator certificate database.

   New or existing certificates saved in the Netscape Communicator certificate database adopt the appropriate database format.

5. Copy the Netscape Communicator certificate database files, `cert7.db` and `key3.db`, that contain your certificates to your `.mcc` directory.

   In Windows, the `cert7.db` and `key3.db` files are located in `C:\ProgramFiles\netscape\Users\`*username*

   On UNIX systems, the `cert7.db` and `key3.db` files are located in your home directory, `/$HOME/.netscape.` `$HOME` is your root directory if you are running Administration Server as root. `$HOME` is your user home directory if you are running Administration Server as a user, for example, `/home/`*username* or `/export/home/`*username.*

   In Windows the `.mcc` directory is located in `C:\WINNT\Profiles\`*username*

   On UNIX systems the `.mcc` directory is located in your home directory. For example, if the Administration Server is running as root, then `.mcc` directory is located in the root directory, `/.mcc`. If Administration Server is running as a user, then .`mcc` is in your user directory, `/home/`*username*`/.mcc` or `/export/home/`*username*`/.mcc`.

The next time you start Console, the Select Certificate window appears. Select a certificate from the pull down menu to continue with an encrypted session in Console.

Chapter 11

Using SNMP to Monitor Servers

You can use the Simple Network Management Protocol (SNMP) to manage your Sun ONE servers. This chapter explains how SNMP works and tells you how to set it up on your network. The chapter contains the following sections:

- SNMP Basics
- Setting Up SNMP on UNIX Systems
- Using a Proxy SNMP Agent on UNIX Systems
- Reconfiguring a Native Agent on UNIX Systems
- Configuring the Master Agent on UNIX Systems
- Starting the Master Agent on UNIX Systems
- Enabling the Subagent on UNIX Systems
- Using the Windows SNMP Service

SNMP Basics

SNMP is a protocol used to exchange data about network activity. It defines a standard method of communication used to manage products from different vendors. This standard allows administrators to remotely manage hardware and software located across their network.

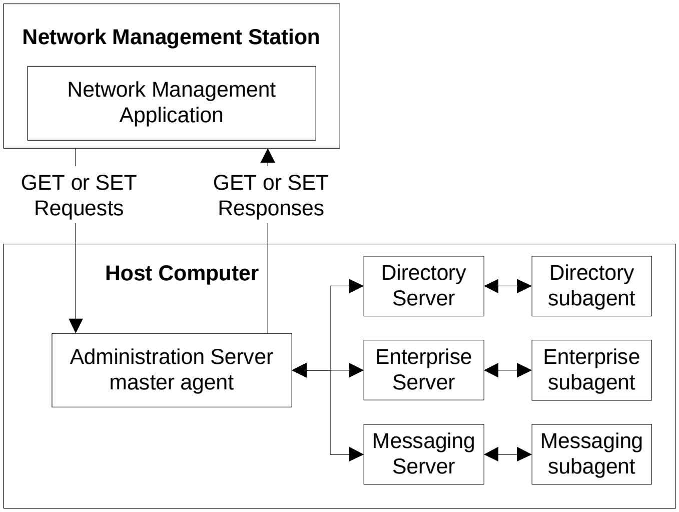
Each piece of controlled hardware and software is known as a managed device. A managed device is anything that runs SNMP, such as a host, router, or Sun ONE server.

The machine used to monitor and configure managed devices is called a network management station. A network management station is usually a powerful workstation running network management applications which graphically show information about managed devices. For example, a network management application might show which servers in your enterprise are running and which are shut down, or the application might report the number and type of error messages received.

Sun ONE servers transmit data to a network management station using two types of agents: SNMP subagents and SNMP master agents. An SNMP subagent gathers information and sends it to an SNMP master agent. The SNMP master agent transfers the data to the network management station. Every Sun ONE server has an SNMP subagent except for Sun ONE Administration Server, which either has a master agent (on UNIX systems) or no agent (on Windows).

A single machine can host multiple subagents, but a machine can only have one master agent. For example, if you have one instance each of Enterprise Server, Directory Server, and Messaging Server installed on one host, each has its own subagent. All three subagents report to the same master agent. This master agent is located on the same host machine as the subagents. Figure 11-1 illustrates this example.

Figure 11-1 Network Management Station and Host Computer



Windows systems offer an SNMP master agent. Sun ONE Administration Server employs this service when utilizing SNMP. You can access and operate this master agent through the Network control panel. In UNIX environments, the master agent is installed with Administration Server.

Some UNIX operating systems support an extended version of SNMP called the SNMP multiplexing protocol (usually known as SMUX). This allows Sun ONE servers to operate without a master agent. For those versions of UNIX that do not support SMUX, you can use Sun ONE Server Console to manage the master agent that Sun ONE provides.

How SNMP Works

A managed device, such as a server, stores its configuration and management settings as variables. Some of these variables can be read and changed over SNMP while others cannot. The variables that the master agent can read and change are called managed objects. Managed objects are defined in a tree-like hierarchy known as a management information base (MIB).

Each Sun ONE server provides a management information base (MIB) for use in SNMP communication. This MIB contains managed objects pertaining to the server's operation. Each managed object has a unique object identifier. A server can report significant events to the network management station by sending "trap" messages (often called just "traps") containing these object identifiers. In addition, the network management station can initiate communication, and then specify one or more object identifiers when querying a server's MIB for data. The network management station can also remotely change variables in the MIB by specifying an object identifier and sending its new value.

Sun ONE MIBs

Each Sun ONE server has its own MIB. All Sun ONE MIBs are located in the *ServerRoot*`/plugins/snmp` directory.

A server's MIB contains variable definitions used when managing that particular server. Some of these variables can be modified over SNMP by a network management station while others are flagged as read-only or inaccessible. See your server documentation for detailed information about its management variables.

The Administration Server MIB

Sun ONE Administration Server stores its MIB in a file called `netscape-main.mib`.

The Administration Server MIB lists the object identifiers for all installed Sun ONE servers. It also defines the object identifier shared by all Sun ONE servers. This object identifier is

```
netscape OJBECT IDENTIFIER: :={enterprises 1450}
```

The `netscape-main.mib` file may look like this:

```
--
-- Netscape Main Mib for SNMP support
--
NETSCAPE-MIB DEFINITIONS ::=
BEGIN
   IMPORTS OBJECT-TYPE
                   FROM SNMPv2-SMI
           MODULE-IDENTITY
                   FROM SNMPv2-SMI
           enterprises
                   FROM ObjectIds
           OBJECT-IDENTITY, Counter64
                  FROM SNMPv2-SMI;

   netscape OBJECT IDENTIFIER ::= { enterprises 1450 }
-- All netscape sub-agents must branch off of the netscape root
-- above. Following  objids for individual sub-agents have been
-- taken already.
-- http    OBJECT IDENTIFIER ::= { netscape 1 }
-- nsmail  OBJECT IDENTIFIER ::= { netscape 5 }
--
END
```

Types of SNMP Messages

SNMP defines three types of messages: GET, SET, and trap. The network management station uses GET messages to request data and SET messages to change variable values in the MIB. The messages sent by a server to the network management station are known as trap messages.

The following examples illustrate how a network management station, and the servers it communicates with, use GET and trap messages.

Network Management Station-Initiated Communication

A network management station can request information from a server or change the value of a variable stored in a server's MIB. For example:

1. The network management station sends a GET message to the Administration Server master agent. The GET message is a request for the number of Directory Server errors encountered since the server was last started.
2. The master agent forwards the message to the Directory Server's SNMP subagent.
3. The subagent retrieves the data.
4. The subagent sends the data to the master agent. The master agent sends a trap message containing the data to the network management station.
5. The network management station displays the data through its network management application.

Server-Initiated Communication

The server subagent sends a trap message to the network management station when a significant event has occurred. For example:

1. The Directory Server's subagent informs the master agent that the server has stopped.
2. The master agent sends a trap message reporting the event to the network management station.
3. The network management station displays the information textually or graphically through its network management application.

Setting Up SNMP on UNIX Systems

In general, to use SNMP on UNIX Systems you must have a master agent and at least one subagent installed and running on your system. You need to install a master agent before you can enable a subagent. Some UNIX systems have their own SNMP master agent. If your system has one of these *native agents*, you can either disable it or change the port number that it uses. If you disable the native agent, you can only use the master agent included with Administration Server. If you change the port number that the native agent uses, you can use it alongside Administration Server's master agent.

The procedures for setting up SNMP are different depending upon your system. Table 11-1 provides an overview of the procedures to follow in various situations. The actual procedures are described in detail later in this chapter.

Before you begin, examine your system.

- Is your system already running an SNMP agent that's native to your operating system?
- If so, does your native SNMP agent support SMUX communication? If your native agent supports SMUX, you don't need to install a master agent. However, you do need to change the native agent's configuration.

If you are unsure of how to verify this information, see your system documentation.

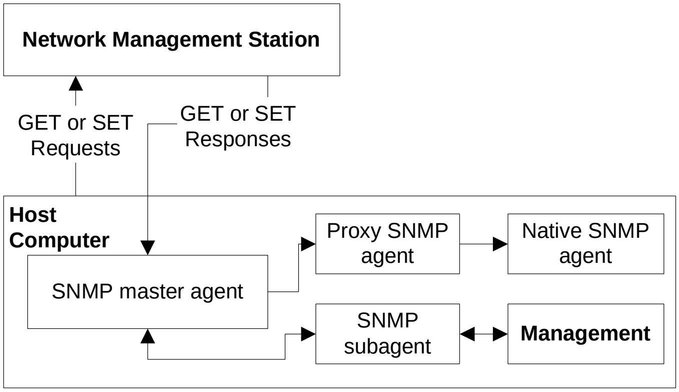
Table 11-1 Overview of Procedures for Enabling SNMP Master Agents and Subagents

If your server meets these conditions...	... follow these procedures
• The system does not have a native agent, or the native agent is not currently running.	1. Start the master agent. 2. Enable the subagent for each server installed on the system.
• The native agent is running, SMUX is not supported, and the system does not need to continue using the native agent.	1. Stop the native agent. 2. Start the master agent. 3. Enable the subagent for each server installed on the system.
• The native agent is running, SMUX is not supported, and the system needs to continue using the native agent.	1. Install and start a proxy SNMP agent. 2. Restart the native agent using a port number that is different from the master agent's port number. 3. Start the master agent. 4. Enable the subagent for each server installed on the system.
• The native agent is running and SMUX is supported.	1. Reconfigure the SNMP native agent. 2. Enable the subagent for each server installed on the system.

Using a Proxy SNMP Agent on UNIX Systems

If you want to use a native agent and the Sun ONE Server Console master agent concurrently, you must set up a proxy agent. The proxy agent fields requests from the Sun ONE master agent and then passes them on to the native agent. This scenario is illustrated in Figure 11-2.

Figure 11-2 Using a Proxy Agent With a Native SNMP Agent



In order to use both master agents simultaneously, you need to install and start the proxy SNMP agent. You also have to restart the native SNMP master agent using a port number other than the one used by the Sun ONE Server Console master agent.

Installing and Starting the Proxy SNMP Agent

Before you install the proxy SNMP agent, make sure to stop the native master agent. See your system documentation for detailed instructions.

To Install the SNMP Proxy Agent

- Edit the `CONFIG` file located in the *ServerRoot*`/plugins/snmp/sagt` directory so that it includes the port that the SNMP proxy agent listens to. The file also needs to include the MIB trees and traps that the SNMP proxy agent forwards.

  Here is a sample `CONFIG` file:

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES 1.3.6.1.2.1.1,
1.3.6.1.2.1.2,
1.3.6.1.2.1.3,
1.3.6.1.2.1.4,
1.3.6.1.2.1.5,
1.3.6.1.2.1.6,
1.3.6.1.2.1.7,
1.3.6.1.2.1.8
FORWARD ALL TRAPS;
```

To Start the SNMP Proxy Agent

- At the command prompt, enter

  `sagt -c CONFIG&`

After the proxy SNMP agent starts, you need to restart the native agent on the port you specified in the `CONFIG` file.

To Restart the Native Agent

- At the command prompt, enter

  `snmpd -P` *portNumber* (specified in the `CONFIG` file)

  For example, on the Solaris platform, using the port in the sample `CONFIG` file above, you would enter

  `snmpd -P 1161`

Reconfiguring a Native Agent on UNIX Systems

If your native agent supports SMUX, you don't need to install a master agent. However, you do need to change the native agent's configuration.

UNIX uses several configuration files to screen its communications. On some systems, `/etc/snmp/conf/snmpd.conf` needs to be changed so that the native agent accepts incoming messages from SMUX subagents. To change the file, add a line defining each subagent by its object identifier.

For example, you might add this line to `snmpd.conf`:

`smux 1.3.6.1.4.1.1.1450.1 ''` *IPAddress netMask*

where *IPAddress* is the IP address of the host on which the subagent is running and *netMask* is the network mask of that host (for example, `255.255.0.0`).

NOTE	Do not use the loopback address 127.0.0.1; use the actual host IP address instead.

Configuring the Master Agent on UNIX Systems

In order to use SNMP, you must configure the master agent by specifying community strings and trap destinations.

Community Strings

A community string is a password text string that an SNMP master agent uses for authorization. Whenever a network management station sends a message, it includes a community string. The agent receiving the message can then verify whether the network management station is authorized to obtain information. Community strings are not concealed when sent in SNMP packets; they are sent as ASCII text.

To ensure that a network management station is authorized to obtain information, the SNMP master agent compares the community string sent by the station to its list of accepted community strings. If the community string is listed, the network management station is authenticated.

Trap Destinations

An SNMP trap is a message the SNMP agent sends to a network management station. For example, an SNMP agent might send a trap when a server goes down. The SNMP agent must know the address of the network management station in order to send traps. This address is called a trap destination.

Configuring the Master Agent using Sun ONE Server Console

Sun ONE Server Console provides an easy way to work with SNMP parameters. You can add, edit, and remove community strings and trap destinations from the Administration Server management window. You can also set the SNMP operations that a particular community string can request, as well as view any trap destinations you have already configured.

To Add, Edit, or Remove a Community String using Sun ONE Server Console

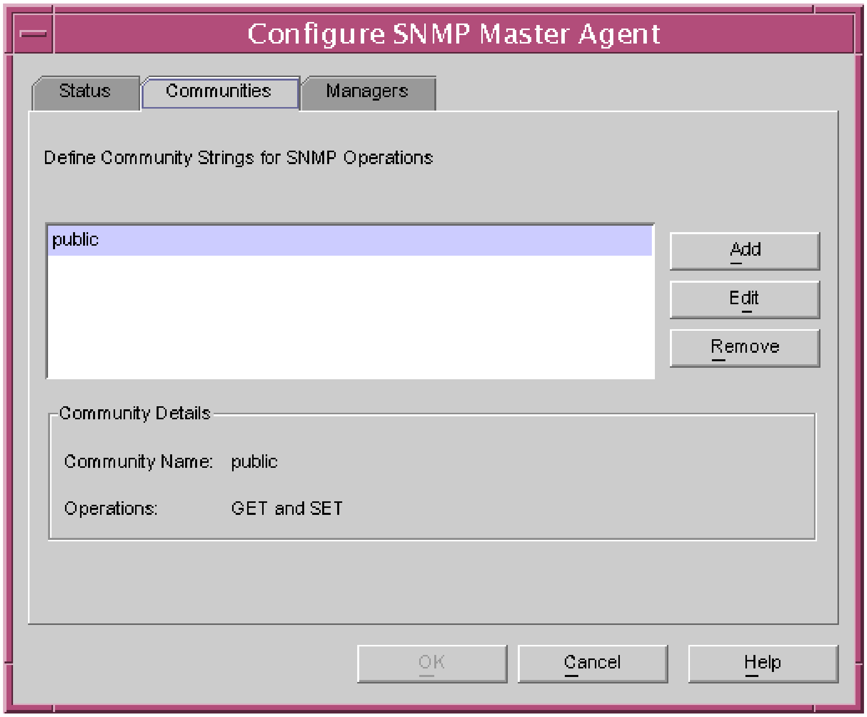
1. In the Sun ONE Server Console navigation tree, select the instance of Administration Server that you want to work with.

2. Click Open to open the management window for the server instance.

3. Click the Tasks tab.

4. Click the Configure SNMP Master Agent button, and then click Communities.

Figure 11-3 The Communities Tab



Configure SNMP Master Agent

Define Community Strings for SNMP Operations

public

Community Details

Community Name: public

Operations: GET and SET

5. Click the appropriate button for the task you are performing.

   - If you want to add a community string, click Add.
   - If you want to edit a community string, select it, and then click Edit.
   - If you want to remove a community string, select it, and then click Remove.

6. Enter community string information as necessary.

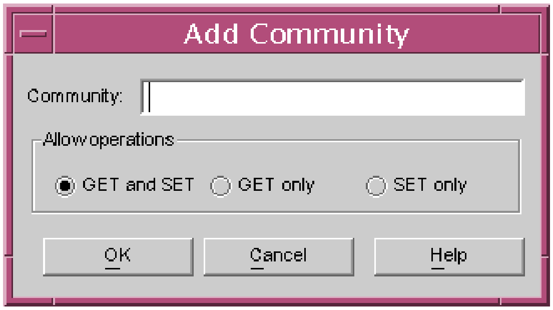
   Community. Enter a community string you want to add, or edit the listed community string.

   GET and SET. Choose this option if you want to use this community string for requesting data, replying to messages, and setting variable values.

   GET only. Choose this option if you want to use this community string only for requesting data and replying to messages.

   SET only. Choose this option if you want to use this community string only for setting variable values.

Figure 11-4 Adding a Community



Add Community

Community:

Allow operations

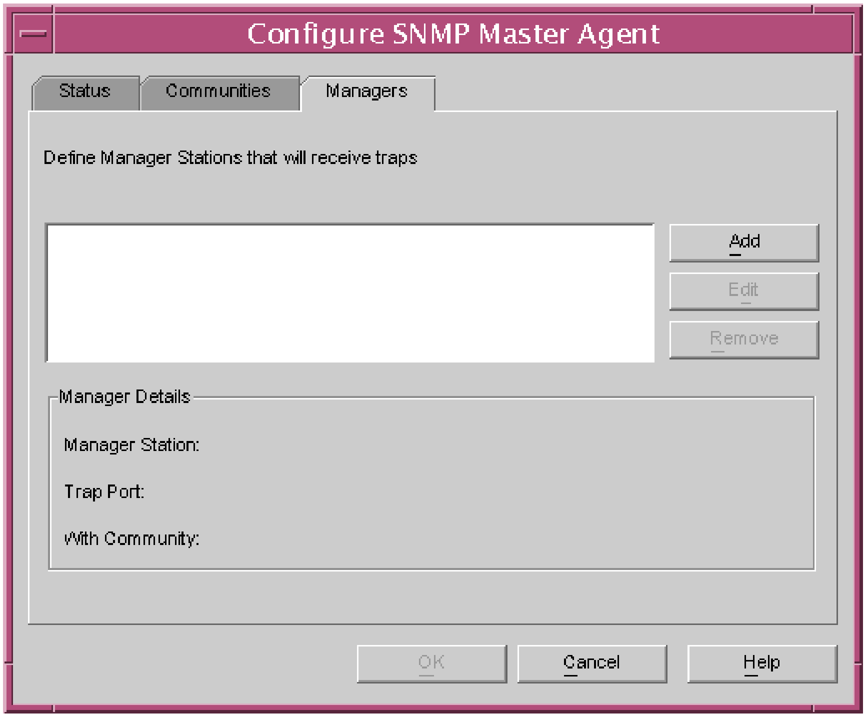
● GET and SET ○ GET only ○ SET only

7. Click OK.

To Add, Edit, or Remove a Trap Destination

1. In the Sun ONE Server Console navigation tree, select the instance of Administration Server on which the master agent is running.
2. Click Open to open the management window for the server instance.
3. Click the Tasks tab.
4. Click the Configure SNMP Master Agent button, then click Managers.

Figure 11-5 The Managers Tab



Define Manager Stations that will receive traps

5. Click the appropriate button for the task you are performing.

    - If you are adding a trap destination, click Add.
    - If you are editing a trap destination, select it, and then click Edit.
    - If you are removing a trap destination, select it, and then click Remove.

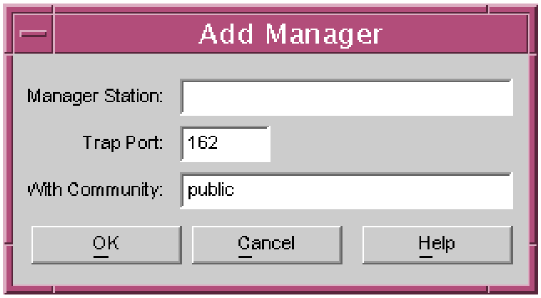
6. If you are adding or editing a trap destination, enter Manager information as necessary:

    Manager Station. Enter a valid system name or an IP address for the network management station.

    Trap Port. Enter the port number that the network management station uses to listen for traps. The default is 162.

    With Community. Enter the community string you want to use in the trap.

Figure 11-6 Adding a Manager



Trap Port: 162

With Community: public

7. Click OK.

Manually Configuring the Master Agent

Although you can easily set SNMP master agent parameters through Sun ONE Server Console, you may want to manually add or modify some settings. You can do this by editing the master agent's configuration file. This file is called `CONFIG` and it contains all master agent settings, whether entered manually or through Sun ONE Server Console.

To Configure the Master SNMP Agent Manually

1. Log in as root.
2. Check to see if there is a native agent (`snmpd`) running on port 161.

   If a native agent is running, make sure you know which MIB trees it supports and how to restart it, then stop it.
3. Edit the `CONFIG` file located in the *ServerRoot*`/plugins/snmp/magt` directory.
4. (Optional) Define `sysContact` and `sysLocation` variables in the `CONFIG` file.

Instructions for editing the `CONFIG` file and defining the `sysContact` and `sysLocation` variables are detailed below.

Editing the Master Agent Config File

The `CONFIG` file defines the community and manager with which the master agent works. The manager value should be a valid system name or an IP address. Here is an example of a basic `CONFIG` file:

```
COMMUNITY           public
                     ALLOW ALL OPERATIONS

MANAGER             your_manager_station_name
                    SEND ALL TRAPS TO PORT 162
                     WITH COMMUNITY public
```

Defining sysContact and sysLocation Variables

You can edit the `CONFIG` file to include initial values for the `sysContact` and `sysLocation` variables (these variables are defined as part of MIB-II, the MIB section of the second version of SNMP). The value for `sysContact` specifies the person in charge of the host system on which the master agent runs. The value for `sysLocation` specifies a physical address where the host machine can be found.

The following example `CONFIG` file defines the `sysContract` and `sysLocation` variables. The strings for the variables in this example are enclosed in quotes. Any string that contains spaces, line breaks, or tabs must be in quotes. Alternatively, you can omit the quotes and specify the value of these whitespace characters in hexadecimal notation.

```
COMMUNITY           public
                     ALLOW ALL OPERATIONS

MANAGER             nms2
                    SEND ALL TRAPS TO PORT 162
                    WITH COMMUNITY public
INITIAL             sysLocation 'Server room
                    4150 Network Circle
                    Santa Clara, CA 95054
                     USA'

INITIAL             sysContact 'John Doe
                     email: <john.doe@sun.com>'
```

Starting the Master Agent on UNIX Systems

Once you have configured the SNMP master agent, you can start it from Sun ONE Server Console or from the command line.

Starting the Agent Using Sun ONE Server Console

Sun ONE Server Console can start the SNMP master agent on the standard port (161) only. If you want to use a non-standard port, see “Starting the Agent From the Command Line” below.

To Start the Master Agent Using Sun ONE Server Console

1. Log in as root.
2. Check to see if there is a native agent (snmpd) running on port 161.

   If a native agent is running, make sure you know which MIB trees it supports and how to restart it, then stop it.
3. In the Sun ONE Server Console navigation tree, select the instance of Administration Server on which the master agent is running.
4. Click Open to open the management window for the server instance.
5. Click the Tasks tab.
6. Double-click Configure SNMP Master Agent.
7. Click the Start button.

Starting the Agent From the Command Line

If you do not want to start the SNMP master agent from Sun ONE Server Console, you can launch it from the command prompt. If you want to run the agent on a port other than 161, you must modify your CONFIG or system services file and then start the agent from the command line.

To Start the Agent on the Standard Port

- Enter the following at the command prompt to start the master agent on port 161:

  ```
  magt CONFIG INIT&
  ```

  The INIT file contains information from the MIB-II system group, including system location and contact information. If INIT doesn’t already exist, starting the master agent for the first time creates it. An invalid manager name in the CONFIG file causes the master agent to fail during startup.

To Start the Agent on a Non-Standard Port Using the Config File

1. In the `CONFIG` file, specify a transport mapping for each interface over which the master agent listens for SNMP requests from network management stations. Transport mappings allow the master agent to accept connections on both the standard port and a nonstandard port.

   The maximum number of concurrent SNMP requests is limited by your target system's limits on the number of open sockets or file descriptors per system process.

   Here is an example of a transport mapping entry:

   ```
   TRANSPORT          extraordinary   SNMP
                      OVER UDP SOCKET
                       AT PORT 1161
   ```

2. After manually editing the `CONFIG` file, you should start the master agent by typing the following at the command prompt:

   ```
   # magt CONFIG INIT&
   ```

To Start the Agent on a Non-Standard Port Using System Services

- Edit the `/etc/services` file to allow the master agent to accept connections on the standard port as well as on a nonstandard port. For information on editing this file, see your system documentation.

Enabling the Subagent on UNIX Systems

For information on enabling the subagent, see the documentation for your Sun ONE server. If you need more information, see your system documentation.

Using the Windows SNMP Service

Windows implements SNMP as a service. Any Sun ONE servers that use SNMP communicate directly with this service. Sun ONE Administration Server does not perform any SNMP-related tasks on Windows. All SNMP-related tasks are handled by the operating system.

To Set Up SNMP on Windows Systems

1. Install the SNMP service on your server.

   Refer to your Windows documentation for instructions.

2. Configure your server software to use SNMP.

   For more information, see your server documentation.

3. Click Start, and then choose Settings > Control Panel.
4. Open the Services control panel.
5. Select the SNMP service from the list of services and then click the Start button.
6. Click Close to exit the Services control panel.

Part 5

Appendixes

Appendix A, “Introduction to Public-Key Cryptography”

Appendix B, “Introduction to SSL”

Appendix A

Introduction to Public-Key Cryptography

Public-key cryptography and related standards and techniques underlie security features of many Sun ONE products, including signed and encrypted mail, form signing, object signing, single sign-on, and the Secure Sockets Layer (SSL) protocol. This appendix introduces the basic concepts of public-key cryptography. This appendix contains the following sections:

- Internet Security Issues
- Encryption and Decryption
- Digital Signatures
- Certificates and Authentication
- Managing Certificates

For an overview of SSL, see Appendix B, “Introduction to SSL.”

Internet Security Issues

All communication over the Internet uses the Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP allows information to be sent from one computer to another through a variety of intermediate computers and separate networks before it reaches its destination.

The great flexibility of TCP/IP has led to its worldwide acceptance as the basic Internet and intranet communications protocol. At the same time, the fact that TCP/IP allows information to pass through intermediate computers makes it possible for a third party to interfere with communications in the following ways:

- **Eavesdropping.** Information remains intact, but its privacy is compromised. For example, someone could learn your credit card number, record a sensitive conversation, or intercept classified information.
- **Tampering.** Information in transit is changed or replaced and then sent on to the recipient. For example, someone could alter an order for goods or change a person's resume.
- **Impersonation.** Information passes to a person who poses as the intended recipient. Impersonation can take two forms, Spoofing and Misrepresentation.
- **Spoofing.** A person pretends to be someone else. For example, a person can pretend to have the mail address `jdoe@example.com`, or a computer can identify itself as a site called `www.example.com` when it is not. This type of impersonation is known as spoofing.
- **Misrepresentation.** A person or organization misrepresents itself. For example, suppose the site `www.mozilla.com` pretends to be a furniture store when it is really just a site that takes credit-card payments but never sends any goods.

Normally, users of the many cooperating computers that make up the Internet or other networks don't monitor or interfere with the network traffic that continuously passes through their machines. However, many sensitive personal and business communications over the Internet require precautions that address the threats listed above. Fortunately, a set of well-established techniques and standards known as public-key cryptography make it relatively easy to take such precautions.

Public-key cryptography facilitates the following tasks:

- Encryption and decryption allow two communicating parties to disguise information they send to each other. The sender encrypts, or scrambles, information before sending it. The receiver decrypts, or unscrambles, the information after receiving it. While in transit, the encrypted information is unintelligible to an intruder.
- Tamper detection allows the recipient of information to verify that it has not been modified in transit. Any attempt to modify data or substitute a false message for a legitimate one is detected.
- Authentication allows the recipient of information to determine its origin—that is, to confirm the sender's identity.
- Nonrepudiation prevents the sender of information from claiming at a later date that the information was never sent.

The sections that follow introduce the concepts of public-key cryptography that underlie these capabilities.

Encryption and Decryption

Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. Decryption is the process of transforming encrypted information so that it is intelligible again. A cryptographic algorithm, also called a cipher, is a mathematical function used for encryption or decryption. In most cases, two related functions are employed, one for encryption and the other for decryption.

With most modern cryptography, the ability to keep encrypted information secret is based not on the cryptographic algorithm, which is widely known, but on a number called a key that must be used with the algorithm to produce an encrypted result or to decrypt previously encrypted information. Decryption with the correct key is simple. Decryption without the correct key is very difficult, and in some cases impossible for all practical purposes.

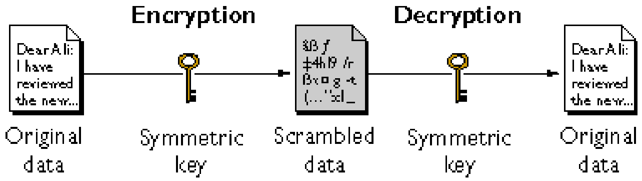
The sections that follow introduce the use of keys for encryption and decryption.

- Symmetric-Key Encryption
- Public-Key Encryption
- Key Length and Encryption Strength

Symmetric-Key Encryption

With symmetric-key encryption, the encryption key can be calculated from the decryption key and vice versa. With most symmetric algorithms, the same key is used for both encryption and decryption.

Figure A-1 Symmetric Key Encryption



Implementations of symmetric-key encryption can be highly efficient, so that users do not experience any significant time delay as a result of the encryption and decryption. Symmetric-key encryption also provides a degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. Thus, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be sure that it is communicating with the other as long as the decrypted messages continue to make sense.

Symmetric-key encryption is effective only if the symmetric key is kept secret by the two parties involved. If anyone else discovers the key, it affects both confidentiality and authentication. A person with an unauthorized symmetric key not only can decrypt messages sent with that key, but can encrypt new messages and send them as if they came from one of the two parties who were originally using the key.

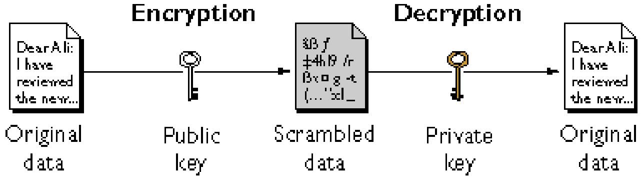
Symmetric-key encryption plays an important role in the SSL protocol, which is widely used for authentication, tamper detection, and encryption over TCP/IP networks. SSL also uses techniques of public-key encryption, which is described in the next section.

Public-Key Encryption

The most commonly used implementations of public-key encryption are based on algorithms patented by RSA Data Security. Therefore, this section describes the RSA approach to public-key encryption.

Public-key encryption (also called asymmetric encryption) involves a pair of keys—a public key and a private key—associated with an entity that needs to authenticate its identity electronically or to sign or encrypt data. Each public key is published, and the corresponding private key is kept secret. (For more information about the way public keys are published, see “Certificates and Authentication,” on page 198.) Data encrypted with your public key can be decrypted only with your private key. Figure A-2 shows a simplified view of the way public-key encryption works.

Figure A-2 Public Key Encryption



The scheme shown in Figure A-2 lets you freely distribute a public key, and only you can read data encrypted using this key. In general, to send encrypted data to someone, you encrypt the data with that person's public key, and the person receiving the encrypted data decrypts it with the corresponding private key.

Compared with symmetric-key encryption, public-key encryption requires more computation and is therefore not always appropriate for large amounts of data. However, it's possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This is the approach used by the SSL protocol.

As it happens, the reverse of the scheme shown in Figure A-2 also works: data encrypted with your private key can be decrypted only with your public key. This would not be a desirable way to encrypt sensitive data, however, because it means that anyone with your public key, which is by definition published, could decrypt the data. Nevertheless, private-key encryption is useful, because it means you can use your private key to sign data with your digital signature—an important requirement for electronic commerce and other commercial applications of cryptography. Client software can then use your public key to confirm that the message was signed with your private key and that it hasn't been tampered with since being signed. "Digital Signatures," on page 196 and subsequent sections describe how this confirmation process works.

Key Length and Encryption Strength

In general, the strength of encryption is related to the difficulty of discovering the key, which in turn depends on both the cipher used and the length of the key. For example, the difficulty of discovering the key for the RSA cipher most commonly used for public-key encryption depends on the difficulty of factoring large numbers, a well-known mathematical problem.

Encryption strength is often described in terms of the size of the keys used to perform the encryption: in general, longer keys provide stronger encryption. Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher. Roughly speaking, 128-bit RC4 encryption is 3 x 10^{26} times stronger than 40-bit RC4 encryption. (For more information about RC4 and other ciphers used with SSL, see Appendix B, "Introduction to SSL.")

Different ciphers may require different key lengths to achieve the same level of encryption strength. The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based. Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values. Thus a 128-bit key for use with a symmetric-key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher. This difference explains why the RSA public-key encryption cipher must use a 512-bit key (or longer) to be considered cryptographically strong, whereas symmetric key ciphers can achieve approximately the same level of strength with a 64-bit key. Even this level of strength may be vulnerable to attacks in the near future.

Digital Signatures

Encryption and decryption address the problem of eavesdropping, one of the three Internet security issues mentioned at the beginning of this appendix. But encryption and decryption, by themselves, do not address the other two problems mentioned in “Internet Security Issues,” on page 191: tampering and impersonation.

This section describes how public-key cryptography addresses the problem of tampering. The sections that follow describe how it addresses the problem of impersonation.

Tamper detection and related authentication techniques rely on a mathematical function called a one-way hash (also called a message digest). A one-way hash is a number of fixed length with the following characteristics:

- The value of the hash is unique for the hashed data. Any change in the data, even deleting or altering a single character, results in a different value.
- The content of the hashed data cannot, for all practical purposes, be deduced from the hash—which is why it is called “one-way.”

As mentioned in “Public-Key Encryption,” on page 194, it’s possible to use your private key for encryption and your public key for decryption. Although this is not desirable when you are encrypting sensitive information, it is a crucial part of digitally signing any data. Instead of encrypting the data itself, the signing software creates a one-way hash of the data, then uses your private key to encrypt the hash. The encrypted hash, along with other information, such as the hashing algorithm, is known as a digital signature.

Figure A-3 shows a simplified view of the way a digital signature can be used to validate the integrity of signed data.

Figure A-3 Digital Signing

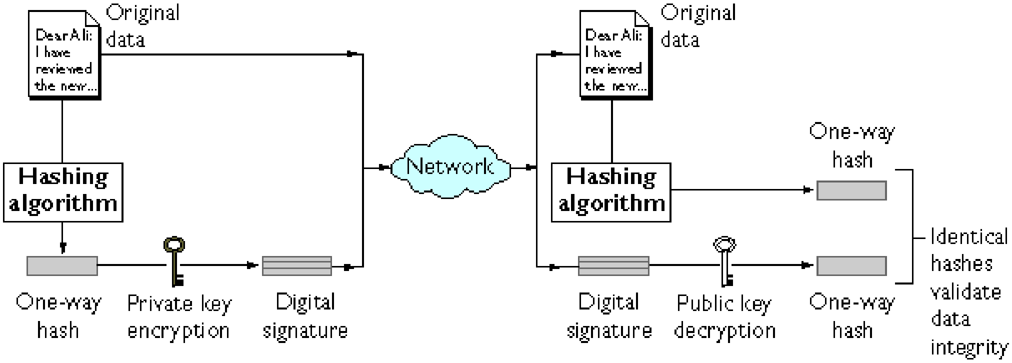


Figure A-3 shows two items transferred to the recipient of some signed data: the original data and the digital signature, which is basically a one-way hash (of the original data) that has been encrypted with the signer's private key. To validate the integrity of the data, the receiving software first uses the signer's public key to decrypt the hash. It then uses the same hashing algorithm that generated the original hash to generate a new one-way hash of the same data. (Information about the hashing algorithm used is sent with the digital signature, although this isn't shown in the figure.) Finally, the receiving software compares the new hash against the original hash. If the two hashes match, the data has not changed since it was signed. If they don't match, the data may have been tampered with since it was signed, or the signature may have been created with a private key that doesn't correspond to the public key presented by the signer.

If the two hashes match, the recipient can be certain that the public key used to decrypt the digital signature corresponds to the private key used to create the digital signature. Confirming the identity of the signer, however, also requires some way of confirming that the public key really belongs to a particular person or other entity. For a discussion of the way this works, see the next section, "Certificates and Authentication," on page 198.

The significance of a digital signature is comparable to the significance of a handwritten signature. Once you have signed some data, it is difficult to deny doing so later—assuming that the private key has not been compromised or out of the owner's control. This quality of digital signatures provides a high degree of nonrepudiation—that is, digital signatures make it difficult for the signer to deny having signed the data. In some situations, a digital signature may be as legally binding as a handwritten signature.

Certificates and Authentication

A Certificate Identifies Someone or Something

A *certificate* is an electronic document used to identify an individual, a server, a company, or some other entity. The certificate also associates that identity with a public key. Like a driver's license, a passport, or other commonly used personal IDs, a certificate provides generally recognized proof of a person's identity. Public-key cryptography uses certificates to address the problem of impersonation (see "Internet Security Issues," on page 191).

To get a driver's license, you typically apply to a government agency, such as the Department of Motor Vehicles, which verifies your identity, your ability to drive, your address, and other information before issuing the license. To get a student ID, you apply to a school or college, which performs different checks (such as whether you have paid your tuition) before issuing the ID. To get a library card, you may need to provide only your name and a utility bill with your address on it.

Certificates work much the same way as any of these familiar forms of identification. Certificate authorities (CAs) are entities that validate identities and issue certificates. They can be either independent third parties or organizations running their own certificate-issuing server software (such as Sun ONE Certificate Management System). The methods used to validate an identity vary depending on the policies of a given CA—just as the methods to validate other forms of identification vary depending on who is issuing the ID and the purpose for which it is used. In general, before issuing a certificate, the CA must use its published verification procedures for that type of certificate to ensure that an entity requesting a certificate is in fact who it claims to be.

The certificate issued by the CA binds a particular public key to the name of the entity the certificate identifies (such as the name of an employee or a server). Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate works with the corresponding private key possessed by the entity identified by the certificate.

In addition to a public key, a certificate always includes the name of the entity it identifies, an expiration date, the name of the CA that issued the certificate, a serial number, and other information. Most importantly, a certificate always includes the digital signature of the issuing CA. The CA's digital signature allows the certificate to function as a "letter of introduction" for users who know and trust the CA but don't know the entity identified by the certificate.

For more information about the role of CAs, see "How CA Certificates Are Used to Establish Trust," on page 211.

Authentication Confirms an Identity

Authentication is the process of confirming an identity. In the context of network interactions, authentication involves the confident identification of one party by another party. Authentication over networks can take many forms. Certificates are one way of supporting authentication.

Network interactions typically take place between a client, such as browser software running on a personal computer, and a server, such as the software and hardware used to host a Web site. *Client authentication* refers to the confident identification of a client by a server (that is, identification of the person assumed to be using the client software). *Server authentication* refers to the confident identification of a server by a client (that is, identification of the organization assumed to be responsible for the server at a particular network address).

Client and server authentication are not the only forms of authentication that certificates support. For example, the digital signature on an email message, combined with the certificate that identifies the sender, provide strong evidence that the person identified by that certificate did indeed send that message. Similarly, a digital signature on an HTML form, combined with a certificate that identifies the signer, can provide evidence, after the fact, that the person identified by that certificate did agree to the contents of the form. In addition to authentication, the digital signature in both cases ensures a degree of nonrepudiation—that is, a digital signature makes it difficult for the signer to claim later not to have sent the email or the form.

Client authentication is an essential element of network security within most intranets or extranets. The sections that follow contrast two forms of client authentication:

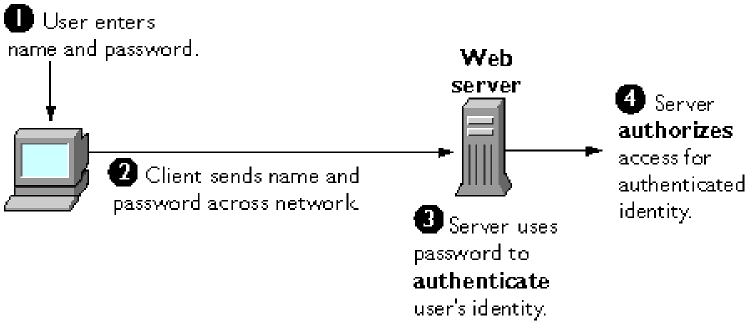
- **Password-Based Authentication.** Almost all server software permits client authentication by means of a name and password. For example, a server might require a user to type a name and password before granting access to the server. The server maintains a list of names and passwords; if a particular name is on the list, and if the user types the correct password, the server grants access.
- **Certificate-Based Authentication.** Client authentication based on certificates is part of the SSL protocol. The client digitally signs a randomly generated piece of data and sends both the certificate and the signed data across the network. The server uses techniques of public-key cryptography to validate the signature and confirm the validity of the certificate.

Password-Based Authentication

Figure A-4 shows the basic steps involved in authenticating a client by means of a name and password. Figure A-4 assumes the following:

- The user has already decided to trust the server, either without authentication or on the basis of server authentication via SSL.
- The user has requested a resource controlled by the server.
- The server requires client authentication before permitting access to the requested resource.

Figure A-4 Using a Password to Authenticate a Client


❶ User enters name and password.
Web server
❷ Client sends name and password across network.
❸ Server uses password to **authenticate** user's identity.
❹ Server **authorizes** access for authenticated identity.

These are the steps shown in Figure A-4:

1. In response to an authentication request from the server, the client displays a dialog box requesting the user’s name and password for that server. The user must supply a name and password separately for each new server the user wishes to use during a work session.

2. The client sends the name and password across the network, either in the clear or over an encrypted SSL connection.
3. The server looks up the name and password in its local password database and, if they match, accepts them as evidence authenticating the user's identity.
4. The server determines whether the identified user is permitted to access the requested resource, and if so allows the client to access it.

With this arrangement, the user must supply a new password for each server, and the administrator must keep track of the name and password for each user, typically on separate servers.

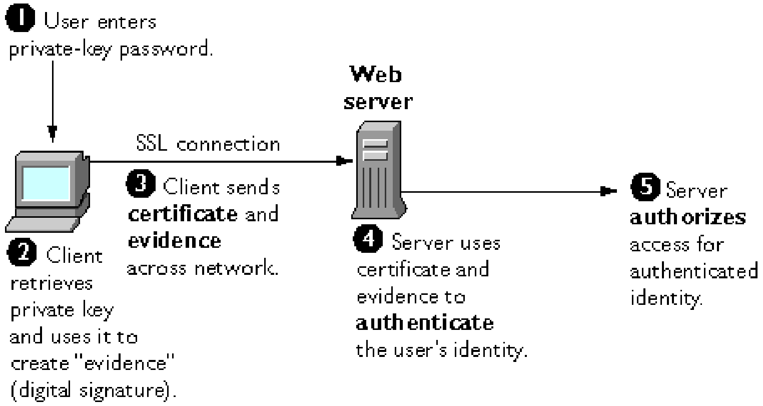
As shown in the next section, one of the advantages of certificate-based authentication is that it can be used to replace the first three steps in Figure A-4 with a mechanism that allows the user to supply just one password (which is not sent across the network) and allows the administrator to control user authentication centrally.

Certificate-Based Authentication

Figure A-5 shows how client authentication works using certificates and the SSL protocol. To authenticate a user to a server, a client digitally signs a randomly generated piece of data and sends both the certificate and the signed data across the network. For the purposes of this discussion, the digital signature associated with some data can be thought of as evidence provided by the client to the server. The server authenticates the user's identity on the strength of this evidence.

Like Figure A-4, Figure A-5 assumes that the user has already decided to trust the server and has requested a resource, and that the server has requested client authentication in the process of evaluating whether to grant access to the requested resource.

Figure A-5 Using a Certificate to Authenticate a Client


1 User enters private-key password.
Web server
SSL connection
3 Client sends **certificate** and **evidence** across network.
5 Server **authorizes** access for authenticated identity.
2 Client retrieves private key and uses it to create "evidence" (digital signature).
4 Server uses certificate and evidence to **authenticate** the user's identity.

Unlike the process shown in Figure A-4, the process shown in Figure A-5 requires the use of SSL. Figure A-5 also assumes that the client has a valid certificate that can be used to identify the client to the server. Certificate-based authentication is generally considered preferable to password-based authentication because it is based on what the user has (the private key) as well as what the user knows (the password that protects the private key). However, it's important to note that these two assumptions are true only if unauthorized personnel have not gained access to the user's machine or password, the password for the client software's private key database has been set, and the software is set up to request the password at reasonably frequent intervals.

NOTE	Neither password-based authentication nor certificate-based authentication address security issues related to physical access to individual machines or passwords. Public-key cryptography can only verify that a private key used to sign some data corresponds to the public key in a certificate. It is the user's responsibility to protect a machine's physical security and to keep the private-key password secret.

These are the steps shown in Figure A-5:

1. The client software maintains a database of the private keys that correspond to the public keys published in any certificates issued for that client. The client asks for the password to this database the first time the client needs to access it during a given session—for example, the first time the user attempts to access an SSL-enabled server that requires certificate-based client authentication. After entering this password once, the user doesn't need to enter it again for the rest of the session, even when accessing other SSL-enabled servers.
2. The client unlocks the private-key database, retrieves the private key for the user's certificate, and uses that private key to digitally sign some data that has been randomly generated for this purpose on the basis of input from both the client and the server. This data and the digital signature constitute "evidence" of the private key's validity. The digital signature can be created only with that private key and can be validated with the corresponding public key against the signed data, which is unique to the SSL session.
3. The client sends both the user's certificate and the evidence (the randomly generated piece of data that has been digitally signed) across the network.
4. The server uses the certificate and the evidence to authenticate the user's identity. (For a detailed discussion of the way this works, see Appendix B, "Introduction to SSL.")
5. At this point the server may optionally perform other authentication tasks, such as checking that the certificate presented by the client is stored in the user's entry in an LDAP directory. The server then continues to evaluate whether the identified user is permitted to access the requested resource. This evaluation process can employ a variety of standard authorization mechanisms, potentially using additional information in an LDAP directory, company databases, and so on. If the result of the evaluation is positive, the server allows the client to access the requested resource.

As you can see by comparing Figure A-5 to Figure A-4, certificates replace the authentication portion of the interaction between the client and the server. Instead of requiring a user to send passwords across the network throughout the day, single sign-on requires the user to enter the private-key database password just once, without sending it across the network. For the rest of the session, the client presents the user's certificate to authenticate the user to each new server it encounters. Existing authorization mechanisms based on the authenticated user identity are not affected.

How Certificates Are Used

- Types of Certificates

- SSL Protocol
- Signed and Encrypted Email
- Form Signing
- Single Sign-On
- Object Signing

Types of Certificates

Five kinds of certificates are commonly used with Sun ONE products:

- **Client SSL certificates.** Used to identify clients to servers via SSL (client authentication). Typically, the identity of the client is assumed to be the same as the identity of a human being, such as an employee in an enterprise. See "Certificate-Based Authentication," on page 201 for a description of the way client SSL certificates are used for client authentication. Client SSL certificates can also be used for form signing and as part of a single sign-on solution.

  Examples: A bank gives a customer a client SSL certificate that allows the bank's servers to identify that customer and authorize access to the customer's accounts. A company might give a new employee a client SSL certificate that allows the company's servers to identify that employee and authorize access to the company's servers.

- **Server SSL certificates.** Used to identify servers to clients via SSL (server authentication). Server authentication may be used with or without client authentication. Server authentication is a requirement for an encrypted SSL session. For more information, see "SSL Protocol," on page 205.

  Example: Internet sites that engage in electronic commerce usually support certificate-based server authentication, at a minimum, to establish an encrypted SSL session and to assure customers that they are dealing with a web site identified with a particular company. The encrypted SSL session ensures that personal information sent over the network, such as credit card numbers, cannot easily be intercepted.

- **S/MIME certificates.** Used for signed and encrypted mail. As with client SSL certificates, the identity of the client is typically assumed to be the same as the identity of a human being, such as an employee in an enterprise. A single certificate may be used as both an S/MIME certificate and an SSL certificate (see "Signed and Encrypted Email," on page 206). S/MIME certificates can also be used for form signing and as part of a single sign-on solution.

Examples: A company deploys combined S/MIME and SSL certificates solely for the purpose of authenticating employee identities, thus permitting signed email and client SSL authentication but not encrypted email. Another company issues S/MIME certificates solely for the purpose of both signing and encrypting email that deals with sensitive financial or legal matters.

- **Object-signing certificates.** Used to identify signers of Java code, JavaScript scripts, or other signed files. For more information, see "Object Signing," on page 208.

  Example: A software company signs software distributed over the Internet to provide users with some assurance that the software is a legitimate product of that company. Using certificates and digital signatures in this manner can also make it possible for users to identify and control the kind of access downloaded software has to their computers.

- **CA certificates.** Used to identify CAs. Client and server software use CA certificates to determine what other certificates can be trusted. For more information, see "How CA Certificates Are Used to Establish Trust," on page 211.

  Example: The CA certificates stored in client software determine what other certificates that client can authenticate. An administrator can implement some aspects of corporate security policies by controlling the CA certificates stored in each user's client.

The sections that follow describes how certificates are used by Sun ONE products.

SSL Protocol

The Secure Sockets Layer (SSL) protocol is a set of rules governing server authentication, client authentication, and encrypted communication between servers and clients. SSL is widely used on the Internet, especially for interactions that involve exchanging confidential information such as credit card numbers.

SSL requires a server SSL certificate, at a minimum. As part of the initial "handshake" process, the server presents its certificate to the client to authenticate the server's identity. The authentication process uses public-key encryption and digital signatures to confirm that the server is in fact the server it claims to be. Once the server has been authenticated, the client and server use techniques of symmetric-key encryption, which is very fast, to encrypt all the information they exchange for the remainder of the session and to detect any tampering that may have occurred.

Servers may optionally be configured to require client authentication as well as server authentication. In this case, after server authentication is successfully completed, the client must also present its certificate to the server to authenticate the client's identity before the encrypted SSL session can be established.

For an overview of client authentication over SSL and how it differs from password-based authentication, see "Authentication Confirms an Identity," on page 199. For more detailed information about SSL, see Appendix B, "Introduction to SSL."

Signed and Encrypted Email

Some mail programs support digitally signed and encrypted mail using a widely accepted protocol known as Secure Multipurpose Internet Mail Extension (S/MIME). Using S/MIME to sign or encrypt mail messages requires the sender of the message to have an S/MIME certificate.

An mail message that includes a digital signature provides some assurance that it was in fact sent by the person whose name appears in the message header, thus providing authentication of the sender. If the digital signature cannot be validated by the mail software on the receiving end, the user is alerted.

The digital signature is unique to the message it accompanies. If the message received differs in any way from the message that was sent—even by the addition or deletion of a comma—the digital signature cannot be validated. Therefore, signed mail also provides some assurance that the mail has not been tampered with. As discussed at the beginning of this appendix, this kind of assurance is known as nonrepudiation. In other words, signed mail makes it very difficult for the sender to deny having sent the message. This is important for many forms of business communication. (For information about the way digital signatures work, see "Digital Signatures," on page 196.)

S/MIME also makes it possible to encrypt email messages. This is also important for some business users. However, using encryption for email requires careful planning. If the recipient of encrypted email messages loses his or her private key and does not have access to a backup copy of the key, for example, the encrypted messages can never be decrypted.

Form Signing

Many kinds of e-commerce require the ability to provide persistent proof that someone has authorized a transaction. Although SSL provides transient client authentication for the duration of an SSL connection, it does not provide persistent authentication for transactions that may occur during that connection. S/MIME provides persistent authentication for mail, but e-commerce often involves filling in a form on a web page rather than sending a mail message.

The Sun ONE technology known as form signing addresses the need for persistent authentication of financial transactions. Form signing allows a user to associate a digital signature with web-based data generated as the result of a transaction, such as a purchase order or other financial document. The private key associated with either a client SSL certificate or an S/MIME certificate may be used for this purpose.

When a user clicks the Submit button on a web-based form that supports form signing, a dialog box appears that displays the exact text to be signed. The form designer can either specify the certificate that should be used or allow the user to select a certificate from among client SSL and S/MIME certificates. When the user clicks OK, the text is signed, and both the text and the digital signature are submitted to the server. The server can then use a Sun ONE utility called the Signature Verification Tool to validate the digital signature.

Single Sign-On

Network users are frequently required to remember multiple passwords for the various services they use. For example, a user might have to type different passwords to log into the network, collect mail, use directory services, use the corporate calendar program, and access various servers. Multiple passwords are an ongoing headache for both users and system administrators. Users have difficulty keeping track of different passwords, tend to choose poor ones, and tend to write them down in obvious places. Administrators must keep track of a separate password database on each server and deal with potential security problems related to the fact that passwords are sent over the network routinely and frequently.

Solving this problem requires some way for a user to log in once, using a single password, and get authenticated access to all network resources that user is authorized to use—without sending any passwords over the network. This capability is known as *single sign-on.*

Both client SSL certificates and S/MIME certificates can play a significant role in a comprehensive single sign-on solution. For example, one form of single sign-on supported by Sun ONE products relies on SSL client authentication (see “Certificate-Based Authentication,” on page 201.). A user can log in once, using a single password to the local client’s private-key database, and get authenticated access to all SSL-enabled servers that user is authorized to use—without sending any passwords over the network. This approach simplifies access for users, because they don’t need to enter passwords for each new server. It also simplifies network management, since administrators can control access by controlling lists of certificate authorities (CAs) rather than much longer lists of users and passwords.

In addition to using certificates, a complete single-sign on solution must address the need to interoperate with enterprise systems, such as the underlying operating system, that rely on passwords or other forms of authentication.

Object Signing

Sun ONE products support a set of tools and technologies called object signing. Object signing uses standard techniques of public-key cryptography to let users get reliable information about code they download in much the same way they can get reliable information about shrink-wrapped software.

Most importantly, object signing helps users and network administrators implement decisions about software distributed over intranets or the Internet—for example, whether to allow Java applets signed by a given entity to use specific computer capabilities on specific users' machines.

The "objects" signed with object signing technology can be applets or other Java code, JavaScript scripts, plug-ins, or any kind of file. The "signature" is a digital signature. Signed objects and their signatures are typically stored in a special file called a JAR file.

Software developers and others who wish to sign files using object-signing technology must first obtain an object-signing certificate.

Contents of a Certificate

The contents of certificates supported by Sun ONE and many other software companies are organized according to the X.509 v3 certificate specification, which has been recommended by the International Telecommunications Union (ITU), an international standards body, since 1988.

Users don't usually need to be concerned about the exact contents of a certificate. However, system administrators working with certificates may need some familiarity with the information provided here.

Distinguished Names

An X.509 v3 certificate binds a distinguished name (DN) to a public key. A DN is a series of name-value pairs, such as `uid=doe`, that uniquely identify an entity—that is, the certificate *subject.*

For example, this might be a typical DN for an employee of Sun Microsystems, Inc.:

```
uid=jdoe,e=john.doe@sun.com,cn=John Doe,dc=sun,dc=com,c=US
```

The abbreviations before each equal sign in this example have these meanings:

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

DNs may include a variety of other name-value pairs. They are used to identify both certificate subjects and entries in directories that support the Lightweight Directory Access Protocol (LDAP).

The rules governing the construction of DNs can be quite complex and are beyond the scope of this appendix. For comprehensive information about DNs, see *A String Representation of Distinguished Names* at the following URL:

`http://www.ietf.org/rfc/rfc1485.txt`

A Typical Certificate

Every X.509 certificate consists of two sections:

The data section includes the following information:

- The version number of the X.509 standard supported by the certificate.
- The certificate's serial number. Every certificate issued by a CA has a serial number that is unique among the certificates issued by that CA.
- Information about the user's public key, including the algorithm used and a representation of the key itself.
- The DN of the CA that issued the certificate.
- The period during which the certificate is valid (for example, between 1:00 p.m. on November 15, 2003 and 1:00 p.m. November 15, 2004)
- The DN of the certificate subject (for example, in a client SSL certificate this would be the user's DN), also called the subject name.
- Optional *certificate extensions*, which may provide additional data used by the client or server. For example, the certificate type extension indicates the type of certificate—that is, whether it is a client SSL certificate, a server SSL certificate, a certificate for signing email, and so on. Certificate extensions can also be used for a variety of other purposes.

The signature section includes the following information:

- The cryptographic algorithm, or cipher, used by the issuing CA to create its own digital signature. For more information about ciphers, see Appendix B, “Introduction to SSL.”
- The CA’s digital signature, obtained by hashing all of the data in the certificate together and encrypting it with the CA's private key.

Here are the data and signature sections of a certificate in human-readable format:

```
Certificate:
Data:
   Version: v3 (0x2)
   Serial Number: 3 (0x3)
   Signature Algorithm: PKCS #1 MD5 With RSA Encryption
   Issuer: OU=Ace Certificate Authority, O=Example Industry, C=US
   Validity:
    Not Before: Fri Oct 17 18:36:25 2003
    Not  After: Sun Oct 17 18:36:25 2004
   Subject: CN=Jane Doe, OU=Finance, O=Example Industry, C=US
   Subject Public Key Info:
    Algorithm: PKCS #1 RSA Encryption
    Public Key:
       Modulus:
          00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86:
          ed:27:40:4d:86:b3:05:c0:01:bb:50:15:c9:de:dc:85:19:22:
          43:7d:45:6d:71:4e:17:3d:f0:36:4b:5b:7f:a8:51:a3:a1:00:
          98:ce:7f:47:50:2c:93:36:7c:01:6e:cb:89:06:41:72:b5:e9:
          73:49:38:76:ef:b6:8f:ac:49:bb:63:0f:9b:ff:16:2a:e3:0e:
          9d:3b:af:ce:9a:3e:48:65:de:96:61:d5:0a:11:2a:a2:80:b0:
          7d:d8:99:cb:0c:99:34:c9:ab:25:06:a8:31:ad:8c:4b:aa:54:
          91:f4:15
       Public Exponent: 65537 (0x10001)
   Extensions:
    Identifier: Certificate Type
      Critical: no

      Certified Usage:
      SSL Client
    Identifier: Authority Key Identifier
      Critical: no
      Key Identifier:
        f2:f2:06:59:90:18:47:51:f5:89:33:5a:31:7a:e6:5c:fb:36:
        26:c9
   Signature:
    Algorithm: PKCS #1 MD5 With RSA Encryption
   Signature:
 6d:23:af:f3:d3:b6:7a:df:90:df:cd:7e:18:6c:01:69:8e:54:65:fc:06:
 30:43:34:d1:63:1f:06:7d:c3:40:a8:2a:82:c1:a4:83:2a:fb:2e:8f:fb:
```

```
 f0:6d:ff:75:a3:78:f7:52:47:46:62:97:1d:d9:c6:11:0a:02:a2:e0:cc:
 2a:75:6c:8b:b6:9b:87:00:7d:7c:84:76:79:ba:f8:b4:d2:62:58:c3:c5:
 b6:c1:43:ac:63:44:42:fd:af:c8:0f:2f:38:85:6d:d6:59:e8:41:42:a5:
 4a:e5:26:38:ff:32:78:a1:38:f1:ed:dc:0d:31:d1:b0:6d:67:e9:46:a8:
 d:c4
```

Here is a certificate displayed in the 64-byte-encoded form interpreted by software:

```
-----BEGIN CERTIFICATE-----
MIICKzCCAZSgAwIBAgIBAzANBgkqhkiG9w0BAQQFADA3MQswCQYDVQQGEwJVUzER
MA8GA1UEChMITmV0c2NhcGUxFTATBgNVBAsTDFN1cHJpeWEncyBDQTAeFw05NzEw
MTgwMTM2MjVaFw05OTEwMTgwMTM2MjVaMEgxCzAJBgNVBAYTAlVTMREwDwYDVQQK
EwhOZXRzY2FwZTENMAsGA1UECxMEUHViczEXMBUGA1UEAxMOU3Vwcml5YSBTaGV0
dHkwgZ8wDQYJKoZIhvcNAQEFBQADgY0AMIGJAoGBAMr6eZiPGfjX3uRJgEjmKiqG
7SdATYazBcABu1AVyd7chRkiQ31FbXFOGD3wNktbf6hRo6EAmM5/R1AskzZ8AW7L
iQZBcrXpc0k4du+2Q6xJu2MPm/8WKuMOnTuvzpo+SGXelmHVChEqooCwfdiZywyZ
NMmrJgaoMa2MS6pUkfQVAgMBAAGjNjA0MBEGCWCGSAGG+EIBAQQEAwIAgDAfBgNV
HSMEGDAWgBTy8gZZkBhHUfWJM1oxeuZc+zYmyTANBgkqhkiG9w0BAQQFAAOBgQBt
I6/z07Z635DfzX4XbAFpjlRl/AYwQzTSYx8GfcNAqCqCwaSDKvsuj/vwbf91o3j3
UkdGYpcd2cYRCgKi4MwqdWyLtpuHAH18hHZ5uvi00mJYw8W2wUOsY0RC/a/IDy84
hW3WWehBUqVK5SY4/zJ4oTjx7dwNMdGwbWfpRqjd1A==
-----END CERTIFICATE-----
```

How CA Certificates Are Used to Establish Trust

Certificate authorities (CAs) are entities that validate identities and issue certificates. They can be either independent third parties or organizations running their own certificate-issuing server software (such as the Sun ONE Certificate Management System).

Any client or server software that supports certificates maintains a collection of trusted CA certificates. These CA certificates determine which other certificates the software can validate—in other words, which issuers of certificates the software can trust. In the simplest case, the software can validate only certificates issued by one of the CAs for which it has a certificate. It's also possible for a trusted CA certificate to be part of a chain of CA certificates, each issued by the CA above it in a certificate hierarchy.

The sections that follow explains how certificate hierarchies and certificate chains determine what certificates software can trust.

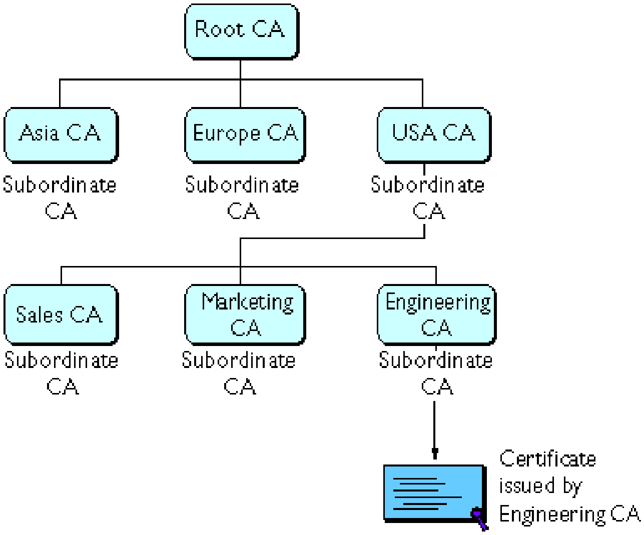
- CA Hierarchies
- Certificate Chains
- Verifying a Certificate Chain

CA Hierarchies

In large organizations, it may be appropriate to delegate the responsibility for issuing certificates to several different certificate authorities. For example, the number of certificates required may be too large for a single CA to maintain; different organizational units may have different policy requirements; or it may be important for a CA to be physically located in the same geographic area as the people to whom it is issuing certificates.

It's possible to delegate certificate-issuing responsibilities to subordinate CAs. The X.509 standard includes a model for setting up a hierarchy of CAs.

Figure A-6 A Hierarchy of Certificate Authorities



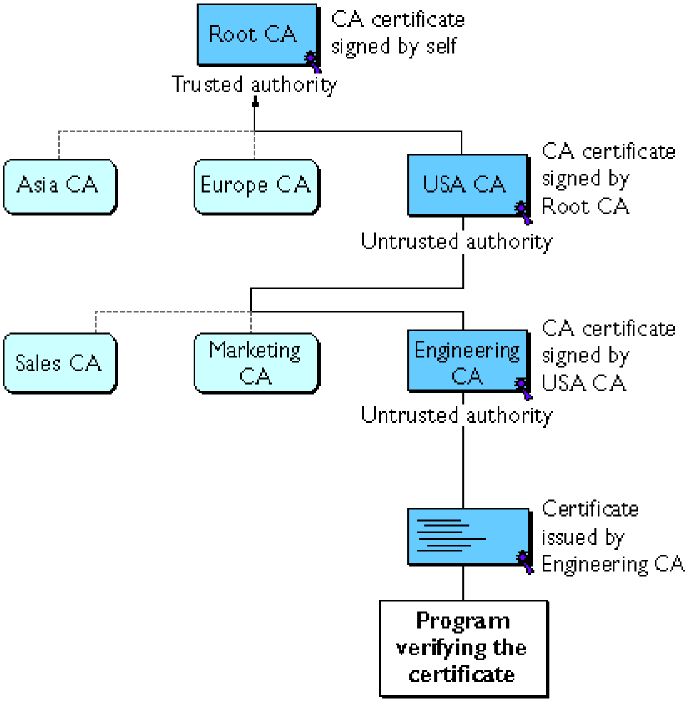
In this model, the root CA is at the top of the hierarchy. The root CA's certificate is a *self-signed certificate:* that is, the certificate is digitally signed by the same entity—the root CA—that the certificate identifies. The CAs that are directly subordinate to the root CA have CA certificates signed by the root CA. CAs under the subordinate CAs in the hierarchy have their CA certificates signed by the higher-level subordinate CAs.

Organizations have a great deal of flexibility in terms of the way they set up their CA hierarchies. Figure A-6 shows just one example; many other arrangements are possible.

Certificate Chains

CA hierarchies are reflected in certificate chains. A *certificate chain* is series of certificates issued by successive CAs. Figure A-7 shows a certificate chain leading from a certificate that identifies some entity through two subordinate CA certificates to the CA certificate for the root CA (based on the CA hierarchy shown in Figure A-6).

Figure A-7 A Certificate Chain


Root CA
CA certificate signed by self
Trusted authority
Asia CA
Europe CA
USA CA
CA certificate signed by Root CA
Untrusted authority
Sales CA
Marketing CA
Engineering CA
CA certificate signed by USA CA
Untrusted authority
Certificate issued by Engineering CA
Program verifying the certificate

A certificate chain traces a path of certificates from a branch in the hierarchy to the root of the hierarchy. In a certificate chain, the following occur:

- Each certificate is followed by the certificate of its issuer.
- Each certificate contains the name (DN) of that certificate's issuer, which is the same as the subject name of the next certificate in the chain.

  In Figure A-7, the Engineering CA certificate contains the DN of the CA (that is, USA CA), that issued that certificate. USA CA's DN is also the subject name of the next certificate in the chain.

- Each certificate is signed with the private key of its issuer. The signature can be verified with the public key in the issuer's certificate, which is the next certificate in the chain.

  In Figure A-7, the public key in the certificate for the USA CA can be used to verify the USA CA's digital signature on the certificate for the Engineering CA.

Verifying a Certificate Chain

Certificate chain verification is the process of making sure a given certificate chain is well-formed, valid, properly signed, and trustworthy. Sun ONE software uses the following procedure for forming and verifying a certificate chain, starting with the certificate being presented for authentication:

1. The certificate validity period is checked against the current time provided by the verifier's system clock.
2. The issuer's certificate is located. The source can be either the verifier's local certificate database (on that client or server) or the certificate chain provided by the subject (for example, over an SSL connection).
3. The certificate signature is verified using the public key in the issuer's certificate.
4. If the issuer's certificate is trusted by the verifier in the verifier's certificate database, verification stops successfully here. Otherwise, the issuer's certificate is checked to make sure it contains the appropriate subordinate CA indication in the Sun ONE certificate type extension, and chain verification returns to step 1 to start again, but with this new certificate. Figure A-8 presents an example of this process.

Figure A-8 Verifying A Certificate Chain

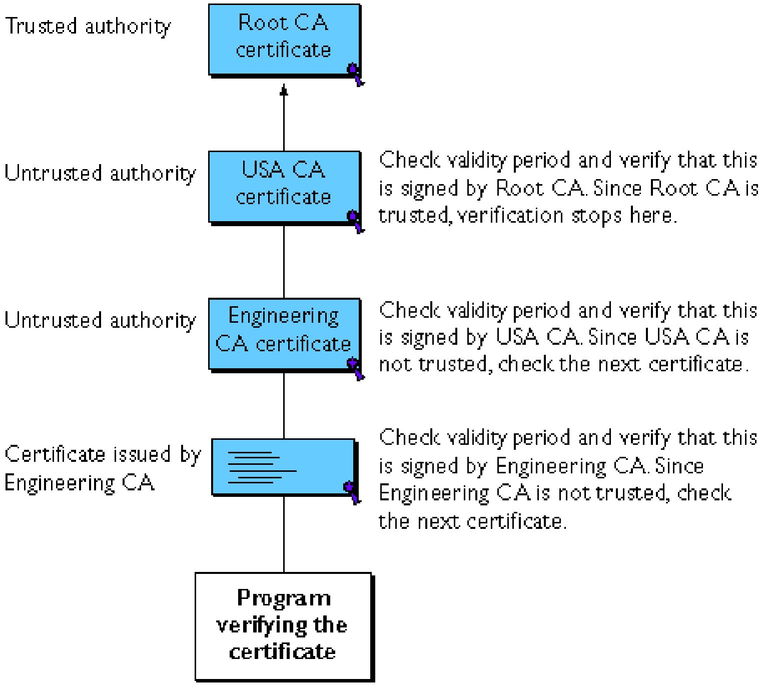
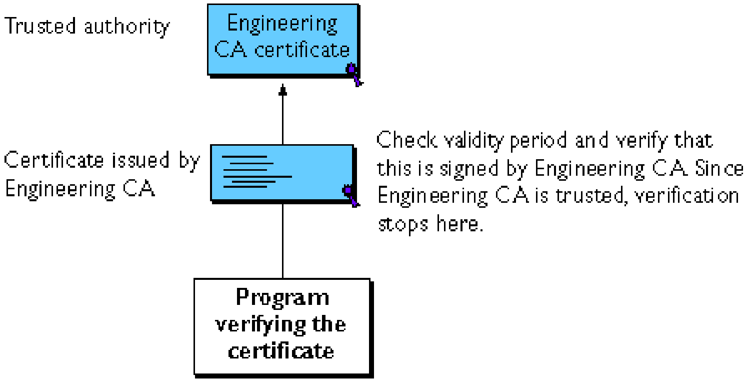

Trusted authority
Root CA certificate
Untrusted authority
USA CA certificate
Check validity period and verify that this is signed by Root CA. Since Root CA is trusted, verification stops here.
Untrusted authority
Engineering CA certificate
Check validity period and verify that this is signed by USA CA. Since USA CA is not trusted, check the next certificate.
Certificate issued by Engineering CA
Check validity period and verify that this is signed by Engineering CA. Since Engineering CA is not trusted, check the next certificate.
Program verifying the certificate

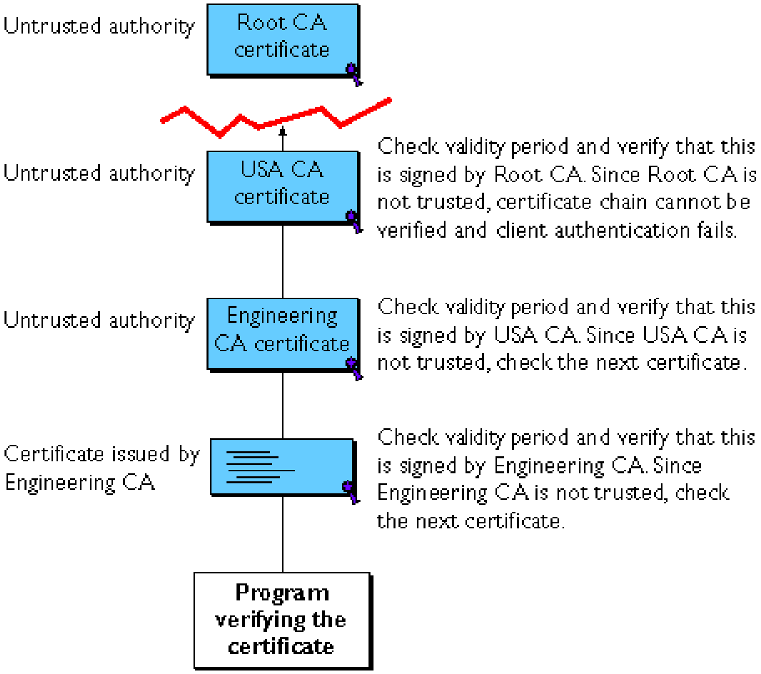
Figure A-8 shows what happens when only Root CA is included in the verifier's local database. If a certificate for one of the intermediate CAs shown in Figure A-8, such as Engineering CA, is found in the verifier's local database, verification stops with that certificate, as shown in Figure A-9.

Figure A-9 Verifying A Certificate Chain to an Intermediate CA


Trusted authority
Engineering CA certificate
Certificate issued by Engineering CA
Check validity period and verify that this is signed by Engineering CA. Since Engineering CA is trusted, verification stops here.
Program verifying the certificate

Expired validity dates, an invalid signature, or the absence of a certificate for the issuing CA at any point in the certificate chain causes authentication to fail. For example, Figure A-10 shows how verification fails if neither the Root CA certificate nor any of the intermediate CA certificates are included in the verifier's local database.

Figure A-10 A Certificate Chain that Cannot Be Verified


Untrusted authority
Root CA certificate
Untrusted authority
USA CA certificate
Check validity period and verify that this is signed by Root CA. Since Root CA is not trusted, certificate chain cannot be verified and client authentication fails.
Untrusted authority
Engineering CA certificate
Check validity period and verify that this is signed by USA CA. Since USA CA is not trusted, check the next certificate.
Certificate issued by Engineering CA
Check validity period and verify that this is signed by Engineering CA. Since Engineering CA is not trusted, check the next certificate.
Program verifying the certificate

For general information about the way digital signatures work, see "Digital Signatures," on page 196. For a more detailed description of the signature verification process in the context of SSL client and server authentication, see Appendix B, "Introduction to SSL."

Managing Certificates

The set of standards and services that facilitate the use of public-key cryptography and X.509 v3 certificates in a network environment is called the *public key infrastructure* (PKI). PKI management is complex topic beyond the scope of this appendix. The sections that follow introduce some of the specific certificate management issues addressed by Sun ONE products.

- Issuing Certificates
- Certificates and the LDAP Directory
- Key Management
- Renewing and Revoking Certificates
- Registration Authorities

Issuing Certificates

The process for issuing a certificate depends on the certificate authority that issues it and the purpose for which it is used. The process for issuing nondigital forms of identification varies in similar ways. For example, if you want to get a generic ID card (not a driver's license) from the Department of Motor Vehicles in California, the requirements are straightforward: you need to present some evidence of your identity, such as a utility bill with your address on it and a student identity card. If you want to get a regular driving license, you also need to take a test—a driving test when you first get the license, and a written test when you renew it. If you want to get a commercial license for an eighteen-wheeler, the requirements are much more stringent. If you live in some other state or country, the requirements for various kinds of licenses differ.

Similarly, different CAs have different procedures for issuing different kinds of certificates. In some cases the only requirement may be your mail address. In other cases, your UNIX or Windows login and password may be sufficient. At the other end of the scale, for certificates that identify people who can authorize large expenditures or make other sensitive decisions, the issuing process may require notarized documents, a background check, and a personal interview.

Depending on an organization's policies, the process of issuing certificates can range from being completely transparent for the user to requiring significant user participation and complex procedures. In general, processes for issuing certificates should be highly flexible, so organizations can tailor them to their changing needs.

Sun ONE Certificate Management System allows an organization to set up its own certificate authority and issue certificates.

Issuing certificates is one of several managements tasks that can be handled by separate Registration Authorities.

Certificates and the LDAP Directory

The Lightweight Directory Access Protocol (LDAP) for accessing directory services supports great flexibility in the management of certificates within an organization. System administrators can store much of the information required to manage certificates in an LDAP-compliant directory. For example, a CA can use information in a directory to prepopulate a certificate with a new employee’s legal name and other information. The CA can leverage directory information in other ways to issue certificates one at a time or in bulk, using a range of different identification techniques depending on the security policies of a given organization. Other routine management tasks, such as key management and renewing and revoking certificates, can be partially or fully automated with the aid of the directory.

Information stored in the directory can also be used with certificates to control access to various network resources by different users or groups. Issuing certificates and other certificate management tasks can thus be an integral part of user and group management.

In general, high-performance directory services are an essential ingredient of any certificate management strategy. Sun ONE Directory Server is fully integrated with Sun ONE Certificate Management System to provide a comprehensive certificate management solution.

Key Management

Before a certificate can be issued, the public key it contains and the corresponding private key must be generated. Sometimes it may be useful to issue a single person one certificate and key pair for signing operations, and another certificate and key pair for encryption operations. Separate signing and encryption certificates make it possible to keep the private signing key on the local machine only, thus providing maximum nonrepudiation, and to back up the private encryption key in some central location where it can be retrieved in case the user loses the original key or leaves the company.

Keys can be generated by client software or generated centrally by the CA and distributed to users via an LDAP directory. There are trade-offs involved in choosing between local and centralized key generation. For example, local key generation provides maximum nonrepudiation, but may involve more participation by the user in the issuing process. Flexible key management capabilities are essential for most organizations.

Key recovery, or the ability to retrieve backups of encryption keys under carefully defined conditions, can be a crucial part of certificate management (depending on how an organization uses certificates). Key recovery schemes usually involve an *m of n* mechanism: for example, *m* of *n* managers within an organization might have to agree, and each contribute a special code or key of their own, before a particular person's encryption key can be recovered. This kind of mechanism ensures that several authorized personnel must agree before an encryption key can be recovered.

Renewing and Revoking Certificates

Like a driver's license, a certificate specifies a period of time during which it is valid. Attempts to use a certificate for authentication before or after its validity period fails. Therefore, mechanisms for managing certificate renewal are essential for any certificate management strategy. For example, an administrator may wish to be notified automatically when a certificate is about to expire, so that an appropriate renewal process can be completed in plenty of time without causing the certificate's subject any inconvenience. The renewal process may involve reusing the same public-private key pair or issuing a new one.

A driver's license can be suspended even if it has not expired—for example, as punishment for a serious driving offense. Similarly, it's sometimes necessary to revoke a certificate before it has expired—for example, if an employee leaves a company or moves to a new job within the company.

Certificate revocation can be handled in several different ways. For some organizations, it may be sufficient to set up servers so that the authentication process includes checking the directory for the presence of the certificate being presented. When an administrator revokes a certificate, the certificate can be automatically removed from the directory, and subsequent authentication attempts with that certificate fails even though the certificate remains valid in every other respect. Another approach involves publishing a certificate revocation list (CRL)—that is, a list of revoked certificates—to the directory at regular intervals and checking the list as part of the authentication process. For some organizations, it may be preferable to check directly with the issuing CA each time a certificate is presented for authentication. This procedure is sometimes called real-time status checking.

Registration Authorities

Interactions between entities identified by certificates (sometimes called end entities) and CAs are an essential part of certificate management. These interactions include operations such as registration for certification, certificate retrieval, certificate renewal, certificate revocation, and key backup and recovery. In general, a CA must be able to authenticate the identities of end entities before responding to the requests. In addition, some requests need to be approved by authorized administrators or managers before being serviced.

As previously discussed, the means used by different CAs to verify an identity before issuing a certificate can vary widely, depending on the organization and the purpose for which the certificate is used. To provide maximum operational flexibility, interactions with end entities can be separated from the other functions of a CA and handled by a separate service called a *Registration Authority* (RA).

An RA acts as a front end to a CA by receiving end entity requests, authenticating them, and forwarding them to the CA. After receiving a response from the CA, the RA notifies the end entity of the results. RAs can be helpful in scaling an PKI across different departments, geographical areas, or other operational units with varying policies and authentication requirements.

Appendix B

Introduction to SSL

This appendix introduces the Secure Sockets Layer (SSL) protocol. SSL has been universally accepted on the World Wide Web for authenticated and encrypted communication between clients and servers. This appendix contains the following sections:

- The SSL Protocol
- Ciphers Used With SSL
- The SSL Handshake

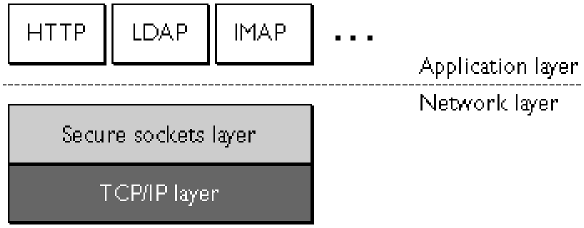
The new Internet Engineering Task Force (IETF) standard protocol called Transport Layer Security (TLS) is based on SSL. The details of the protocol are available in RFC 2246, *The TLS Protocol Version 1.0*. Some Sun ONE products already support TLS.

This appendix is primarily intended for administrators of Sun ONE server products, but the information it contains may also be useful for developers of applications that support SSL. The appendix assumes that you are familiar with the basic concepts of public-key cryptography, as summarized in Appendix A, "Introduction to Public-Key Cryptography."

The SSL Protocol

The Transmission Control Protocol/Internet Protocol (TCP/IP) governs the transport and routing of data over the Internet. Other protocols, such as the HyperText Transport Protocol (HTTP), Lightweight Directory Access Protocol (LDAP), or Internet Messaging Access Protocol (IMAP), run "on top of" TCP/IP in the sense that they all use TCP/IP to support typical application tasks such as displaying web pages or running mail servers.

Figure B-1 Where SSL Runs



The SSL protocol runs above TCP/IP and below higher-level protocols such as HTTP or IMAP. It uses TCP/IP on behalf of the higher-level protocols, and in the process allows an SSL-enabled server to authenticate itself to an SSL-enabled client, allows the client to authenticate itself to the server, and allows both machines to establish an encrypted connection.

These capabilities address fundamental concerns about communication over the Internet and other TCP/IP networks:

- SSL server authentication allows a user to confirm a server's identity. SSL-enabled client software can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs. This confirmation might be important if the user, for example, is sending a credit card number over the network and wants to check the receiving server's identity.

- SSL client authentication allows a server to confirm a user's identity. Using the same techniques as those used for server authentication, SSL-enabled server software can check that a client's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the server's list of trusted CAs. This confirmation might be important if the server, for example, is a bank sending confidential financial information to a customer and wants to check the recipient's identity.

- An encrypted SSL connection requires all information sent between a client and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality. Confidentiality is important for both parties to any private transaction. In addition, all data sent over an encrypted SSL connection is protected with a mechanism for detecting tampering—that is, for automatically determining whether the data has been altered in transit.

The SSL protocol includes two sub-protocols: the SSL record protocol and the SSL handshake protocol. The SSL record protocol defines the format used to transmit data. The SSL handshake protocol involves using the SSL record protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when they first establish an SSL connection. This exchange of messages is designed to facilitate the following actions:

- Authenticate the server to the client.
- Allow the client and server to select the cryptographic algorithms, or ciphers, that they both support.
- Optionally authenticate the client to the server.
- Use public-key encryption techniques to generate shared secrets.
- Establish an encrypted SSL connection.

For more information about the handshake process, see “The SSL Handshake,” on page 226.

Ciphers Used With SSL

The SSL protocol supports the use of a variety of different cryptographic algorithms, or ciphers, for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. Clients and servers may support different cipher suites, or sets of ciphers, depending on factors such as the version of SSL they support, and company policies regarding acceptable encryption strength. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suites they use to authenticate each other, to transmit certificates, and to establish session keys.

Key-exchange algorithms like KEA and RSA key exchange govern the way in which the server and client determine the symmetric keys they both use during an SSL session. The most commonly used SSL cipher suites use RSA key exchange.

The SSL 2.0 and SSL 3.0 protocols support overlapping sets of cipher suites. Administrators can enable or disable any of the supported cipher suites for both clients and servers. When a particular client and server exchange information during the SSL handshake, they identify the strongest enabled cipher suites they have in common and use those for the SSL session.

Decisions about which cipher suites a particular organization decides to enable depend on trade-offs among the sensitivity of the data involved, the speed of the cipher, and the applicability of export rules.

NOTE Sun ONE Server Console does not support all of the cipher suites supported by various clients and servers. To ensure that Sun ONE Server Console can control an SSL-enabled server, the server must enable at least one of the cipher suites for SSL 3.0:

These are the cipher suites for SSL 3.0:

- RC4 with 128-bit encryption and MD5 message authentication
- RC4 with 40-bit encryption and MD5 message authentication
- RC2 with 40-bit encryption and MD5 message authentication
- No encryption, MD5 message authentication only

See Table B-1 for details.

Cipher Suites With RSA Key Exchange

Table B-1 lists the cipher suites supported by SSL that use the Rivest Shamir Aldeman (RSA) key-exchange algorithm. Unless otherwise indicated, all ciphers listed in the table are supported by both SSL 2.0 and SSL 3.0.Cipher suites are listed from strongest to weakest.

The list is not exhaustive.

Table B-1 Cipher Suites Supported by the SSL Protocol That Use the RSA Key-Exchange Algorithm

Strength Category and Recommended Use	Cipher Suites
Strongest Cipher Suite This cipher suite is appropriate for banks and other institutions that handle highly sensitive data. Sun ONE Server Console does not support this cipher suite.	**Triple DES With 168-Bit Encryption and SHA-1 Message Authentication** Triple Data Encryption Standard (Triple DES) is the strongest cipher supported by SSL, but it is not as fast as RC4. Triple DES uses a key three times as long as the key for standard DES. Because the key size is so large, there are more possible keys than for any other cipher—approximately 3.7 * 10^{50}. This cipher suite is Federal Information Processing Standard (FIPS) compliant. Both SSL 2.0 and SSL 3.0 support this cipher suite.

Table B-1 Cipher Suites Supported by the SSL Protocol That Use the RSA Key-Exchange Algorithm

Strength Category and Recommended Use	Cipher Suites
Strong Cipher Suites These cipher suites support encryption that is strong enough for most business or government needs.	**RC4 With 128-Bit Encryption and MD5 Message Authentication** Because the RC4 and RC2 ciphers have 128-bit encryption, they are the second strongest next to Triple DES with 168-bit encryption. RC4 and RC2 128-bit encryption permits approximately 3.4 * 10^{38} possible keys, making them very difficult to crack. RC4 ciphers are the fastest of the supported ciphers. Both SSL 2.0 and SSL 3.0 support this cipher suite. Sun ONE Server Console supports only the SSL 3.0 version of this cipher suite. It does not support the RC2 version of this suite. **DES With 56-Bit Encryption and SHA-1 Message Authentication** DES is stronger than 40-bit encryption, but not as strong as 128-bit encryption. DES 56-bit encryption permits approximately 7.2 * 10^{16} possible keys. This cipher suite is FIPS-compliant. Both SSL 2.0 and SSL 3.0 support this cipher suite, except that SSL 2.0 uses MD5 rather than SHA-1 for message authentication. Sun ONE Server Console does not support this cipher suite.
Less Strong Cipher Suites These cipher suites are not as strong as those listed above, but are widely used.[1]	**RC4 With 40-Bit Encryption and MD5 Message Authentication** RC4 40-bit encryption permits approximately 1.1 * 10^{12} (a trillion) possible keys. RC4 ciphers are the fastest of the supported ciphers. Both SSL 2.0 and SSL 3.0 support this cipher. Sun ONE Server Console supports only the SSL 3.0 version of this cipher suite. **RC2 With 40-Bit Encryption and MD5 Message Authentication** RC2 40-bit encryption permits approximately 1.1 * 10^{12} (a trillion) possible keys. RC2 ciphers are slower than the RC4 ciphers. Both SSL 2.0 and SSL 3.0 support this cipher. Sun ONE Server Console supports only the SSL 3.0 version of this cipher suite.

Table B-1 Cipher Suites Supported by the SSL Protocol That Use the RSA Key-Exchange Algorithm

Strength Category and Recommended Use	Cipher Suites
Weakest Cipher Suite	**No Encryption, MD5 Message Authentication Only**
This cipher suite provides authentication and tamper detection but no encryption. Server administrators must therefore be careful about enabling it, because data sent using this cipher suite is not encrypted and may be accessed by eavesdroppers.	This cipher suite uses MD5 message authentication to detect tampering. It is typically supported in case a client and server have none of the other ciphers in common. This cipher suite is supported by SSL 3.0 but not by SSL 2.0.

[1] Note that for RC4 and RC2 ciphers, the phrase "40-bit encryption" means the keys are still 128 bits long, but only 40 bits have cryptographic significance.

The SSL Handshake

The SSL protocol uses a combination of public-key and symmetric key encryption. Symmetric key encryption is much faster than public-key encryption, but public-key encryption provides better authentication techniques. An SSL session always begins with an exchange of messages called the *SSL handshake*. The handshake allows the server to authenticate itself to the client using public-key techniques, then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. Optionally, the handshake also allows the client to authenticate itself to the server.

The exact programmatic details of the messages exchanged during the SSL handshake are beyond the scope of this appendix. However, the steps involved can be summarized as follows (assuming the use of the cipher suites listed in "Cipher Suites With RSA Key Exchange," on page 224):

1. The client sends the server the client's SSL version number, cipher settings, randomly generated data, and other information the server needs to communicate with the client using SSL.
2. The server sends the client the server's SSL version number, cipher settings, randomly generated data, and other information the client needs to communicate with the server over SSL. The server also sends its own certificate and, if the client is requesting a server resource that requires client authentication, requests the client's certificate.

3. The client uses some of the information sent by the server to authenticate the server (for details, see “Server Authentication,” on page 228). If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client goes on to Step 4.

4. Using all data generated in the handshake so far, the client (with the cooperation of the server, depending on the cipher being used) creates the premaster secret for the session, encrypts it with the server’s public key (obtained from the server’s certificate, sent in Step 2), and sends the encrypted premaster secret to the server.

5. If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case the client sends both the signed data and the client’s own certificate to the server along with the encrypted premaster secret.

6. If the server has requested client authentication, the server attempts to authenticate the client (for details, see “Client Authentication,” on page 231). If the client cannot be authenticated, the session is terminated. If the client can be successfully authenticated, the server uses its private key to decrypt the premaster secret, then performs a series of steps (which the client also performs, starting from the same premaster secret) to generate the master secret.

7. Both the client and the server use the master secret to generate the *session keys,* which are symmetric keys used to encrypt and decrypt information exchanged during the SSL session and to verify its integrity—that is, to detect changes in the data between the time it was sent and the time it is received over the SSL connection.

8. The client sends a message to the server informing it that future messages from the client are encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished.

9. The server sends a message to the client informing it that future messages from the server are encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished.

10. The SSL handshake is now complete, and the SSL session has begun. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity.

Before continuing with the session, Sun ONE servers can be configured to check that the client's certificate is present in the user's entry in an LDAP directory. This configuration option provides one way of ensuring that the client's certificate has not been revoked.

It's important to note that both client and server authentication involve encrypting some piece of data with one key of a public-private key pair and decrypting it with the other key:

- In the case of server authentication, the client encrypts the premaster secret with the server's public key. Only the corresponding private key can correctly decrypt the secret, so the client has some assurance that the identity associated with the public key is in fact the server with which the client is connected. Otherwise, the server cannot decrypt the premaster secret and cannot generate the symmetric keys required for the session, and the session is terminated.
- In the case of client authentication, the client encrypts some random data with the client's private key—that is, it creates a digital signature. The public key in the client's certificate can correctly validate the digital signature only if the corresponding private key was used. Otherwise, the server cannot validate the digital signature and the session is terminated.

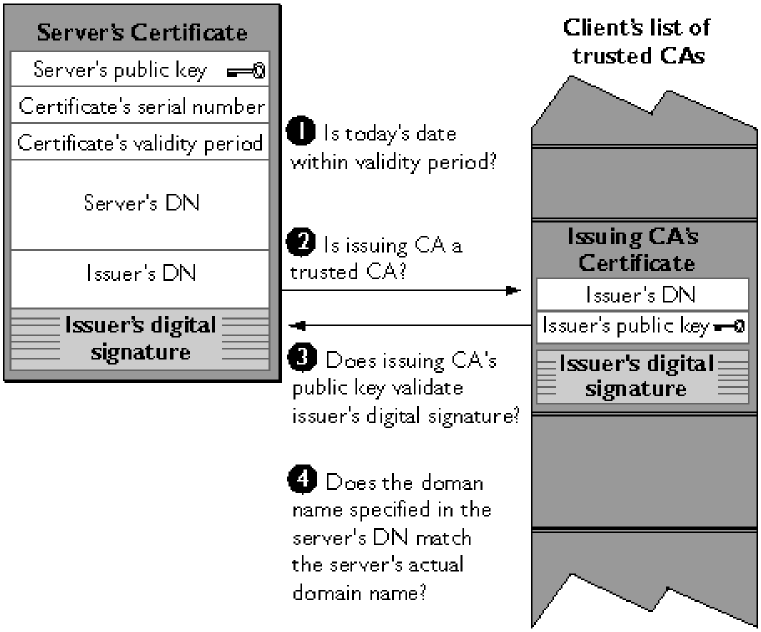
The sections that follow provide more details on server authentication and client authentication.

Server Authentication

SSL-enabled client software always requires server authentication, or cryptographic validation by a client of the server's identity. As explained in Step 2 of "The SSL Handshake," on page 226, the server sends the client a certificate to authenticate itself. The client uses the certificate in Step 3 to authenticate the identity the certificate claims to represent.

To authenticate the binding between a public key and the server identified by the certificate that contains the public key, an SSL-enabled client must receive a "yes" answer to the four questions shown in Figure B-2.

Figure B-2 Authenticating a Client Certificate


Server's Certificate
Server's public key
Certificate's serial number
Certificate's validity period
Server's DN
Issuer's DN
Issuer's digital signature
Client's list of trusted CAs
1 Is today's date within validity period?
2 Is issuing CA a trusted CA?
3 Does issuing CA's public key validate issuer's digital signature?
4 Does the doman name specified in the server's DN match the server's actual domain name?
Issuing CA's Certificate
Issuer's DN
Issuer's public key
Issuer's digital signature

An SSL-enabled client goes through these steps to authenticate a server's identity:

1. **Is today's date within the validity period?** The client checks the server certificate's validity period. If the current date and time are outside of that range, the authentication process won't go any further. If the current date and time are within the certificate's validity period, the client goes on to Step 2.

2. **Is the issuing CA a trusted CA?** Each SSL-enabled client maintains a list of trusted CA certificates, represented by the shaded area on the right side of Figure B-3. This list determines which server certificates the client accepts. If the distinguished name (DN) of the issuing CA matches the DN of a CA on the client's list of trusted CAs, the answer to this question is yes, and the client goes on to Step 3. If the issuing CA is not on the list, the server is not authenticated unless the client can verify a certificate chain ending in a CA that is on the list (see "CA Hierarchies," on page 212 for details).

3. **Does the issuing CA's public key validate the issuer's digital signature?** The client uses the public key from the CA's certificate (which it found in its list of trusted CAs in step 2) to validate the CA's digital signature on the server certificate being presented. If the information in the server certificate has changed since it was signed by the CA or if the CA certificate's public key doesn't correspond to the private key used by the CA to sign the server

certificate, the client won't authenticate the server's identity. If the CA's digital signature can be validated, the server treats the user's certificate as a valid "letter of introduction" from that CA and proceeds. At this point, the client has determined that the server certificate is valid. It is the client's responsibility to take Step 4 before Step 5.

4. **Does the domain name in the server's certificate match the domain name of the server itself?** This step confirms that the server is actually located at the same network address specified by the domain name in the server certificate. Although step 4 is not technically part of the SSL protocol, it provides the only protection against a form of security attack known as "man in the middle." Clients must perform this step and must refuse to authenticate the server or establish a connection if the domain names don't match. If the server's actual domain name matches the domain name in the server certificate, the client goes on to Step 5.

5. **The server is authenticated.** The client proceeds with the SSL handshake. If the client doesn't get to step 5 for any reason, the server identified by the certificate cannot be authenticated, and the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server requires client authentication, the server performs the steps described in "Client Authentication," on page 231.

After the steps described here, the server must successfully use its private key to decrypt the premaster secret the client sends in Step 4 of "The SSL Handshake," on page 226. Otherwise, the SSL session is terminated. This provides additional assurance that the identity associated with the public key in the server's certificate is in fact the server with which the client is connected.

Man-in-the-Middle Attack

The "man in the middle" is a rogue program that intercepts all communication between the client and a server with which the client is attempting to communicate via SSL. The rogue program intercepts the legitimate keys that are passed back and forth during the SSL handshake, substitutes its own, and makes it appear to the client that it is the server, and to the server that it is the client.

The encrypted information exchanged at the beginning of the SSL handshake is actually encrypted with the rogue program's public key or private key, rather than the client's or server's real keys. The rogue program ends up establishing one set of session keys for use with the real server, and a different set of session keys for use with the client. This allows the rogue program not only to read all the data that flows between the client and the real server, but also to change the data without being deleted. Therefore, it is extremely important for the client to check that the domain name in the server certificate corresponds to the domain name of the server with which a client is attempting to communicate—in addition to checking the validity of the certificate by performing the other steps described in "Server Authentication," on page 228.

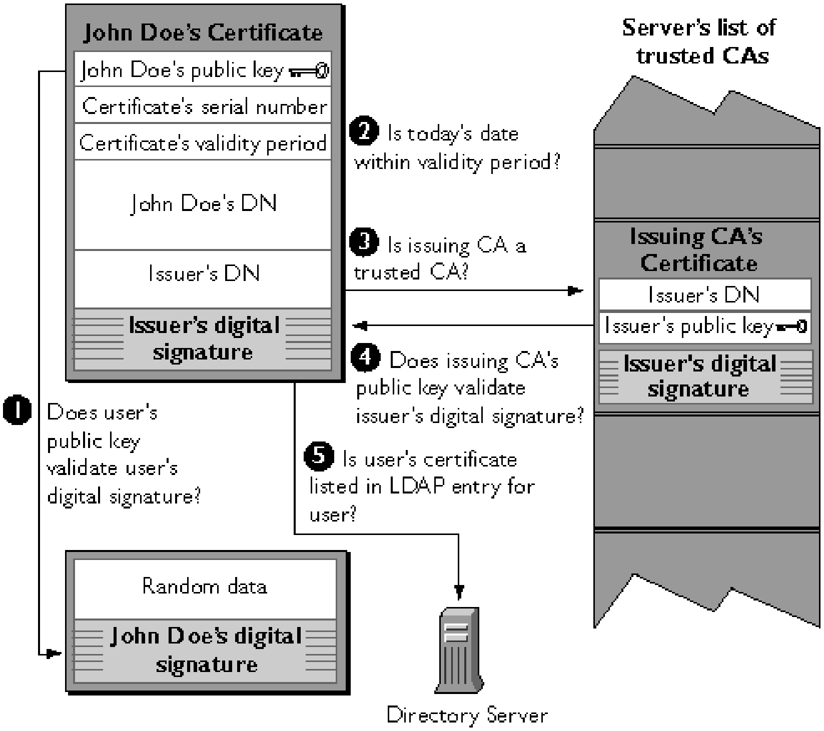
Client Authentication

SSL-enabled servers can be configured to require client authentication, or cryptographic validation by the server of the client's identity. When a server configured this way requests client authentication (see Step 6 of "The SSL Handshake," on page 226), the client sends the server both a certificate and a separate piece of digitally signed data to authenticate itself. The server uses the digitally signed data to validate the public key in the certificate and to authenticate the identity the certificate claims to represent.

The SSL protocol requires the client to create a digital signature by creating a one-way hash from data generated randomly during the handshake and known only to the client and server. The hash of the data is then encrypted with the private key that corresponds to the public key in the certificate being presented to the server.

To authenticate the binding between the public key and the person or other entity identified by the certificate that contains the public key, an SSL-enabled server must receive a "yes" answer to the first four questions shown in Figure B-3. Although the fifth question is not part of the SSL protocol, Sun ONE servers can be configured to support this requirement to take advantage of the user's entry in an LDAP directory as part of the authentication process.

Figure B-3 Authentication and Verification


John Doe's Certificate
John Doe's public key
Certificate's serial number
Certificate's validity period
John Doe's DN
Issuer's DN
Issuer's digital signature
Server's list of trusted CAs
2 Is today's date within validity period?
3 Is issuing CA a trusted CA?
Issuing CA's Certificate
Issuer's DN
Issuer's public key
Issuer's digital signature
4 Does issuing CA's public key validate issuer's digital signature?
1 Does user's public key validate user's digital signature?
5 Is user's certificate listed in LDAP entry for user?
Random data
John Doe's digital signature
Directory Server

An SSL-enabled server goes through these steps to authenticate a user's identity:

1. **Does the user's public key validate the user's digital signature?** The server checks that the user's digital signature can be validated with the public key in the certificate. If so, the server has established that the public key asserted to belong to John Doe matches the private key used to create the signature and that the data has not been tampered with since it was signed.

   At this point, however, the binding between the public key and the DN specified in the certificate has not yet been established. The certificate might have been created by someone attempting to impersonate the user. To validate the binding between the public key and the DN, the server must also complete Step 3 and Step 4.

2. **Is today's date within the validity period?** The server checks the certificate's validity period. If the current date and time are outside of that range, the authentication process won't go any further. If the current date and time are within the certificate's validity period, the server goes on to Step 3.

3. **Is the issuing CA a trusted CA?** Each SSL-enabled server maintains a list of trusted CA certificates, represented by the shaded area on the right side of Figure B-3. This list determines which certificates the server accepts. If the DN of the issuing CA matches the DN of a CA on the server's list of trusted CAs, the answer to this question is yes, and the server goes on to Step 4. If the issuing CA is not on the list, the client is not authenticated unless the server can verify a certificate chain ending in a CA that is on the list (see "CA Hierarchies," on page 212 for details). Administrators can control which certificates are trusted or not trusted within their organizations by controlling the lists of CA certificates maintained by clients and servers.

4. **Does the issuing CA's public key validate the issuer's digital signature?** The server uses the public key from the CA's certificate (which it found in its list of trusted CAs in Step 3) to validate the CA's digital signature on the certificate being presented. If the information in the certificate has changed since it was signed by the CA or if the public key in the CA certificate doesn't correspond to the private key used by the CA to sign the certificate, the server won't authenticate the user's identity. If the CA's digital signature can be validated, the server treats the user's certificate as a valid "letter of introduction" from that CA and proceeds. At this point, the SSL protocol allows the server to consider the client authenticated and proceed with the connection as described in Step 6. Sun ONE servers may optionally be configured to perform Step 5 before Step 6.

5. **Is the user's certificate listed in the LDAP entry for the user?** This optional step provides one way for a system administrator to revoke a user's certificate even if it passes the tests in all the other steps. The Sun ONE Certificate Management System can automatically remove a revoked certificate from the user's entry in the LDAP directory. All servers that are set up to perform this step then refuses to authenticate that certificate or establish a connection. If the user's certificate in the directory is identical to the user's certificate presented in the SSL handshake, the server goes on to step 6.

6. **Is the authenticated client authorized to access the requested resources?** The server checks what resources the client is permitted to access according to the server's access control lists (ACLs) and establishes a connection with appropriate access. If the server doesn't get to step 6 for any reason, the user identified by the certificate cannot be authenticated, and the user is not allowed to access any server resources that require authentication.

Index