Managing Servers with Netscape Console

Version 4.1

Recycled and Recyclable Paper

The Team:
Engineering:Glen Beasley, Atul Bhandari, Andy Hakim, Miodrag Kekic, Terence Kwan, Peter Lee, Steve Pariso, Adam Prishtina, Chih-Ming Shih, David Tompkins, Rob Weltman, Tony Xue
Marketing:Kevin Tsurutome
Publications:Gina Cariaga
Quality Assurance:Sudesh Chandra, Jeesun Cho, Jun Tong

Version 4.1

Part Number 151-06256-00

Printed in the United States of America. 00 99 98 5 4 3 2 1

Netscape Communications Corporation, 501 East Middlefield Road, Mountain View, CA 94043

Contents

Part 1 Overview of Netscape Console

Part 2 Netscape Server Basics

Part 3 Administrator's Guide to Netscape Administration Server

Part 4 Appendixes

Introduction

Managing Servers with Netscape Console provides background information system architects and administrators need to successfully install and manage a Netscape enterprise. Read about Netscape server basics here before you begin actually installing and configuring servers in your enterprise.

What's in This Guide

This book provides information you need to use Netscape servers:

- Part I: Overview of the Netscape Console 4.1
- Part II: Netscape Server Basics
- Part III: Administrator's Guide to the Administration Server

Conventions Used in This Guide

`Monospaced font` This typeface is used for any text that appears on the computer screen or text that you should type. It's also used for file and path names and functions.

Boldface Boldface type is used for window elements such as input areas and check boxes.

Italic Italic type is used for emphasis, book titles, and glossary terms.

Note Sidebar text marks important information. Make sure you read the information before continuing with a task.

[ ] Square brackets enclose commands that are optional. That is, you can omit any text that appears in square brackets.

< > Angle brackets enclose variables. When following examples, replace the angle brackets and their text with text that applies to your situation. For example, when path names appear in angle brackets, substitute the path names used on your computer.

/ Forward slash is used to separate directories in a path. If you use the NT operating system, you might be more familiar with \ in paths, but NT supports both forward and back slashes.

Unix Marks text that applies only to the Unix versions of the administration server.

NT Marks text that applies only to the Windows NT versions of the administration server.

Viewing This Book Online

For your convenience, this book is also available online. When using any Netscape server software, you can view the online version of *Managing Servers with Netscape Console*.

To view the online book:

1. From the Help menu, choose Contents.
2. Click Bookshelf.
3. In the Bookshelf listing, click *Managing Servers with Netscape Console*.

1

Overview of Netscape Console

Chapter

1

Introducing Netscape Console

Netscape Console is a powerful server management tool that uses a graphical interface. Working together with Netscape Directory Server and Administration Server, these three core elements let you view and access all the servers under your control from one central location. You can log in from any system connected to your network to manage a remote server or to make changes in a centralized directory.

This chapter contains the following sections:

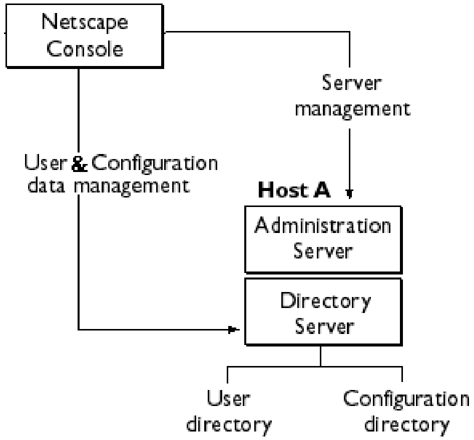
- The Directory Server
- The Administration Server
- Netscape Console

The Directory Server

The Directory Server stores user data and server configuration data used by other servers in the enterprise. User and group entries are contained in one subtree, called the *user directory*, of the Directory Server. Application and server configuration information is stored in another subtree, called the *configuration directory*.

Note When you install Netscape Directory Server 4.0, the Administration Server and Netscape Console are automatically installed for you. You cannot install any other Netscape 4.0 server until you've installed a Directory Server.

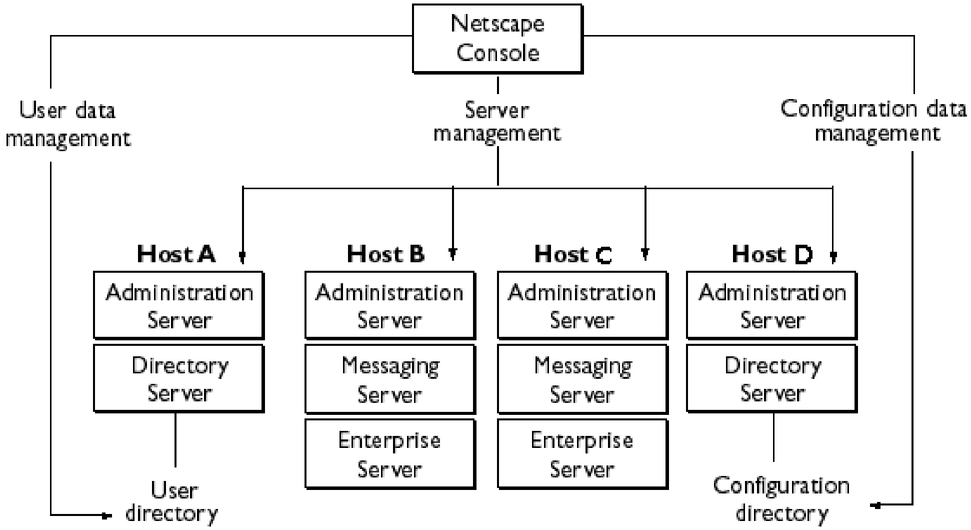
Figure 1.1 Directory Server stores configuration and user data in separate subtrees.



The architecture is flexible and provides a wide range of deployment options. A configuration directory and a user directory can be located in the same Directory Server, or they can be located in physically separate Directory Servers. For example, in small deployments managed by a small number of administrators, it may be suitable to store both configuration and user data in the same Directory Server. In a larger deployment that requires highly delegated user management, it may be more practical to store configuration data and user data in physically separate Directory Servers.

Whenever you install a Netscape 4.0 server, you must specify the location of the configuration directory, and the location of the user directory.

Figure 1.2 In this example, all servers in the enterprise share the same user directory stored on Host A. All servers share the same configuration data stored on Host D.



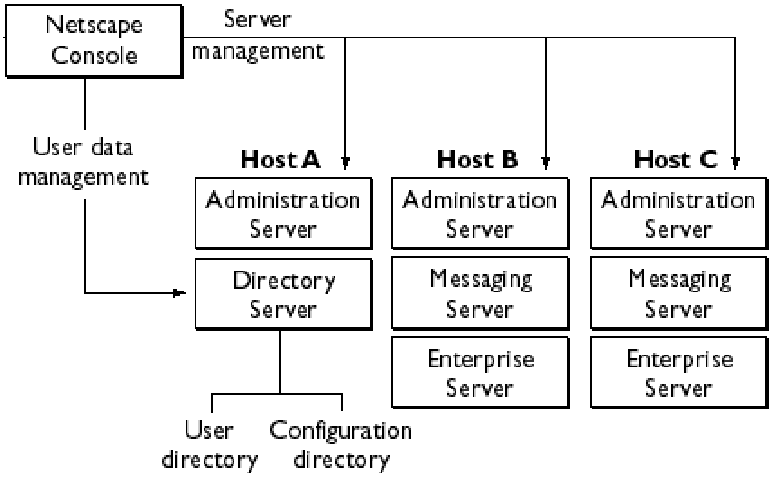
For more information about the configuration and user directories, see “Directory Settings” on page 139.

The Administration Server

The Administration Server is a lightweight HTTP server that acts as the back end to Netscape Console. The Administration Server manages operation requests from all servers installed in a server root or *server group*, and invokes CGI programs to actually perform the requested operations. For example, you can use Netscape Console to change the port number of a Messaging Server. The Console sends the request to the appropriate Administration Server. The Administration Server then invokes the programs that actually change the Messaging Server's port number.

Whenever you install a Netscape 4.0 server, if an Administration Server is not already installed for the server group, one will be automatically installed for you. On Unix, there can be more than one server group installed on a host.

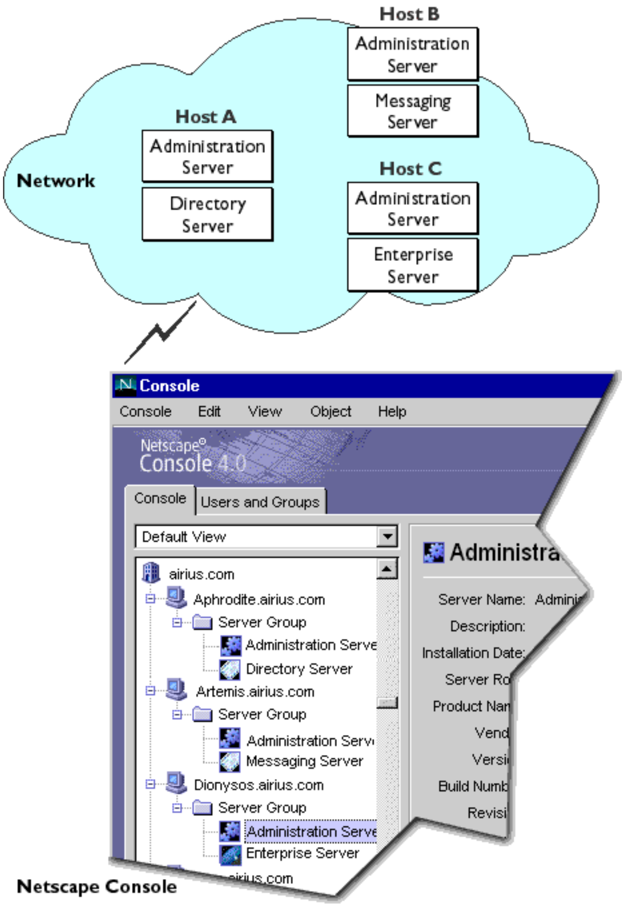
Figure 1.3 Each server group requires an Administration Server. In this example, multiple servers share a single Directory Server



Netscape Console

Netscape Console is a stand-alone Java application. It finds all resources, and applications registered in the Directory Server, and displays them in a graphical interface. Netscape Console functions independently of any server, and you can use it from any computer or workstation connected to your enterprise.

Figure 1.4 Netscape Console graphical interface provides access to all resources under your control.



Chapter

2

The Netscape Server Products Setup Program

This chapter provides an overview of the Netscape Server Products Setup Program and information for using it in various situations.

This chapter contains the following sections:

- It's More Than Just a Setup Program
- Silent Installation
- Installing Netscape Console as a Stand-Alone Application

Note Each Netscape server has its own detailed installation instructions. You'll find these in your server's `Install.htm` file at `http://netscape.com/eng/server`

It's More Than Just a Setup Program

The Setup Program integrates the installation of multiple Netscape servers into a single operation. Use the Netscape Server Products Setup Program each time you need to

- install a new server or server component
- update a server

- install Netscape Console as a stand-alone application

Installing a New Server

Note Each Netscape server has its own detailed installation instructions. Look for the server's `Install.htm` file at `http://netscape.com/eng/server.` This section provides an overview of installation dependencies and options common to all Netscape 4.0 servers.

Directory Server Must Be Installed First

The Setup Program authenticates against an installed Directory Server. During installation, you'll need to provide the Configuration Administrator's ID, password, and base DN. These are stored in the Directory Server. If authentication fails, you won't be able to complete the installation. For detailed information, see your server's `Install.htm` file at `http://netscape.com/eng/server`.

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

Administration Server Is Required in Each Server Root

All Netscape servers require that an Administration Server be installed in each server root. If an Administration Server is not currently installed, the Setup Program automatically installs one for you.

Installation Modes

The Setup Program offers three installation modes: Express, Typical, and Custom.

Express

Use this to get the system running quickly, using default settings a much as possible. This mode was designed for administrators who want to test basic server operation on a particular system before actually deploying the server. It automatically generates administrator names, passwords, and other information required to complete the most basic installation.

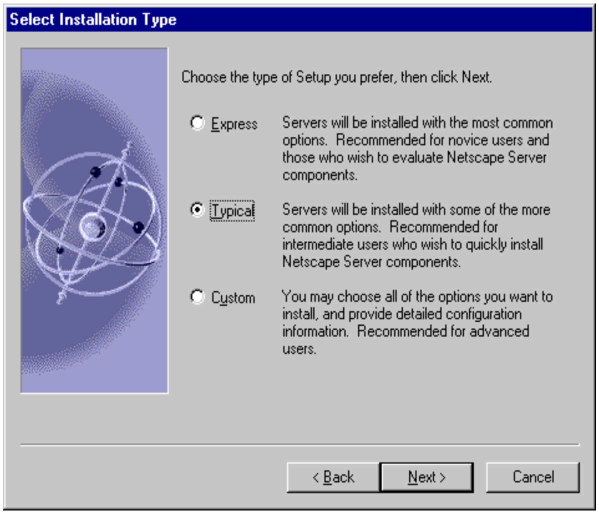
Typical

Use this mode if you want to be able to specify some, but not all, installation options. Administrators use this mode most often because it lets them modify some settings such as directory location, user names, and passwords.

Custom

Use this mode only if you’ve 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.

Figure 2.1 Installation Modes



Select Installation Type

Choose the type of Setup you prefer, then click Next.

○ Express — Servers will be installed with the most common options. Recommended for novice users and those who wish to evaluate Netscape Server components.

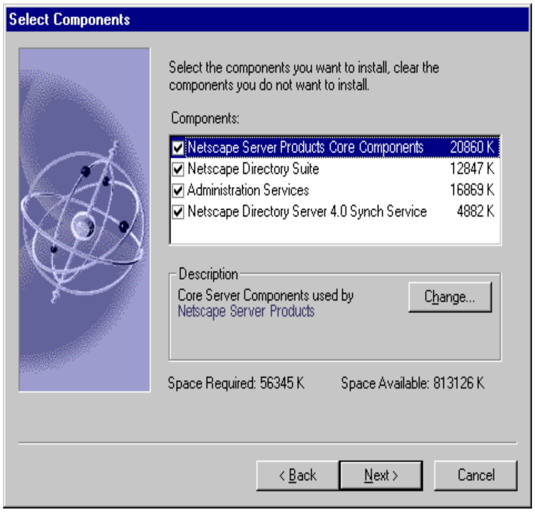
● Typical — Servers will be installed with some of the more common options. Recommended for intermediate users who wish to quickly install Netscape Server components.

○ Custom — You may choose all of the options you want to install, and provide detailed configuration information. Recommended for advanced users.

Silent Installation

The Silent Installation feature allows you to use a file to predefine all the answers that you would normally supply to the Setup Program interactively. This is useful when you want to install a large number of the same type of Netscape server using the same installation specifications. For detailed information on Silent Install, see your server's `Install.htm` file at `http://netscape.com/eng/server`.

Figure 2.2 The Setup Program displays the servers and components available to you.



Select Components

Select the components you want to install, clear the components you do not want to install.

Components:

☑ Netscape Server Products Core Components	20860 K
☑ Netscape Directory Suite	12847 K
☑ Administration Services	16869 K
☑ Netscape Directory Server 4.0 Synch Service	4882 K

Description

Core Server Components used by Netscape Server Products

Space Required: 56345 K   Space Available: 813126 K

Installing Netscape Console as a Stand-Alone Application

You can install Netscape Console as a stand-alone application--without installing a server--on a machine local to you. This is useful when you want to manage servers on remote machines. For detailed information, see your server's `Install.htm` file at `http://netscape.com/eng/server`.

2

Netscape Server Basics

Chapter

3

Using Netscape Console

Netscape Console let's you view all servers under your control. It's where you go to open and manage individual servers.

This chapter contains the following sections:

- Logging In to Netscape Console
- The Navigation Tree
- Customizing Your View of Netscape Console

Logging In to Netscape Console

Netscape Console is a stand-alone Java application. When you start Netscape Console, you connect to an Administration Server in your network. Typically, you log in using your own user name and password.

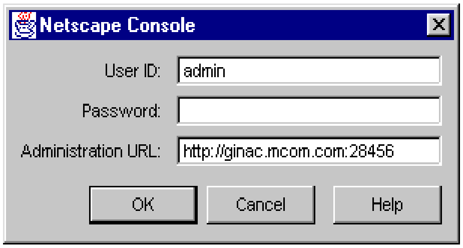
To start Netscape Console and log in:

1. Start Netscape Console:

   Unix. In the server root, enter`./startconsole`.

   Windows NT. From the Start Menu, choose Programs. Then, from the Netscape Server Family Program Group, choose Netscape Console 4.1.

2. In Netscape Console login window, enter your user name, password, and the URL for the Administration Server you want to access.


Netscape Console

User ID: admin

Password:

Administration URL: http://ginac.mcom.com:28456

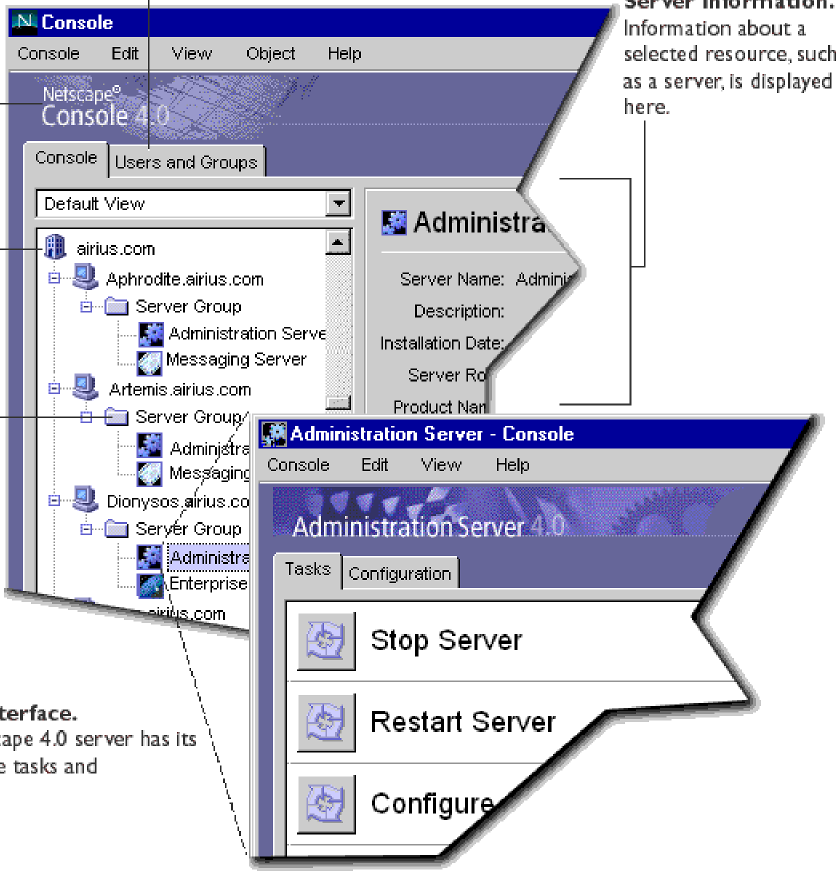
3. Click OK.

The user name and password you use to log in determine which servers and server operations you can access in Netscape Console. See “Network Resources and Administrative Privileges” on page 92 for more information.

The Navigation Tree

The navigation tree represents all the *resources*, or objects, in a Netscape *topology*. A topology includes all resources registered in the same configuration directory. An *administration domain* is a collection of host systems and servers that share the same user directory. A *server group* consists of all servers managed by the same Administration Server. Individual *servers* are the products that provide specific services such as directory database, messaging, and publishing.

Figure 3.1 The navigation tree represents the Netscape topology which includes all resources registered in the same configuration directory. Click the plus (+) or minus (-) signs to expand or collapse a section of the tree.



Banner. Tells you you are in the Netscape Console. Each server also has its own banner.

Users and Groups. Click here to access the user directory.

Server Information. Information about a selected resource, such as a server, is displayed here.

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

Server Group. Includes all servers installed in the same "server root" directory.

Server Interface. Each Netscape 4.0 server has its own unique tasks and features.

Opening a Netscape Server

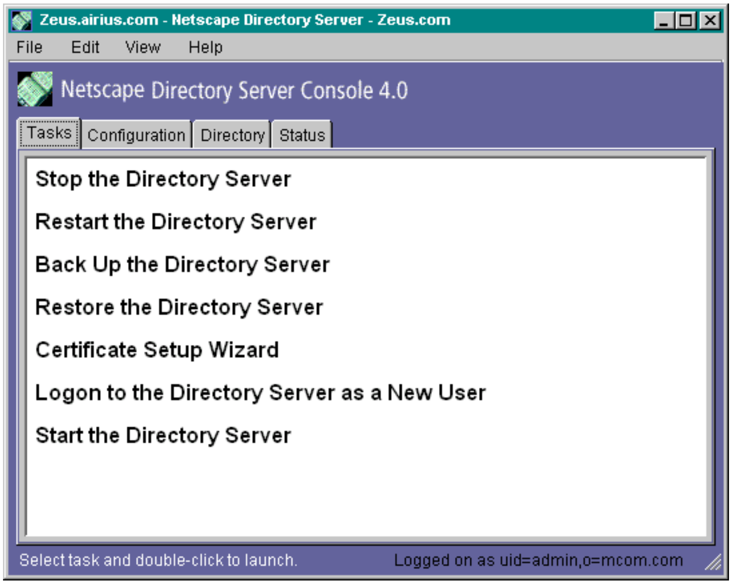
Each Netscape Server has its own set of tasks and configuration settings.

To open a Netscape server:

1. In the navigation tree of Netscape Console, click a server to select it.
2. In the server information section of Netscape Console, click Open.

Each Netscape server has specialized tabs for setting configurations or viewing server-specific information. For detailed information, see the server's *Administrator's Guide.*

Figure 3.2 The Directory Server graphical interface.


Zeus.airius.com - Netscape Directory Server - Zeus.com
File
Edit
View
Help
Netscape Directory Server Console 4.0
Tasks
Configuration
Directory
Status

Stop the Directory Server

Restart the Directory Server

Back Up the Directory Server

Restore the Directory Server

Certificate Setup Wizard

Logon to the Directory Server as a New User

Start the Directory Server

Select task and double-click to launch.

Logged on as uid=admin,o=mcom.com

The Administration Domain

An administration domain is a group of Netscape servers that all use the same user directory for user data management and authentication. For example, you might want to create a separate domain for each division in your company with each domain including all host computers used only by that division.

Creating an Administration Domain

Before you can create a new Administration Domain, you must be member of the Configuration Administrators group.

To create a new administration domain:

1. In Netscape Console, from the Console menu, choose Create Administration Domain.
2. In the Create Administration Domain dialog box, enter domain information:

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

   Description. (Optional) You can enter any string that helps you identify this domain.

   User Directory. Specify the location of the user directory using the fully qualified domain name and port number.

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

   User Directory Subtree. enter the location of the new user directory. Example: o=mcom.com

   Bind DN. Enter the distinguished name for a user who has access permisions to the new user directory. Example: `uid=ginac, ou=people, o=Airius.com.`

   Bind Password. Enter the password of the user above.
3. Click OK.

If you've changed the User Directory Host, User Directory Port, or SSL option, you must restart the server before the change takes effect.

Modifying an Administration Domain

Note Changing these settings will have serious and far-reaching impacts on the rest of the servers in the domain! If you make changes here, you must restart all the servers in the domain.

To modify an administration domain:

1. In Netscape Console, select the domain you want to modify, then click Edit.

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) You can enter any 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. For authentication purposes, you can enter more than one user directory location separated by spaces.

   Example:
   `Eros.Airius.com:389 Zeus.Airius.com:389`

   See "User Authentication and Directory Failover Support" on page 142 for more information.

   `Important` If you specify more than one host computer in this field, each one must be configured identically for each of these settings:

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

   User Directory Subtree. Enter the location of the new user directory. Example: o=mcom.com

   Bind DN. Enter the distinguished name for a user who has access permisions to the new user directory. Example: `uid=ginac, ou=people, o=Airius.com.`

   Bind Password. Enter the password of the user above.

3. Click OK.

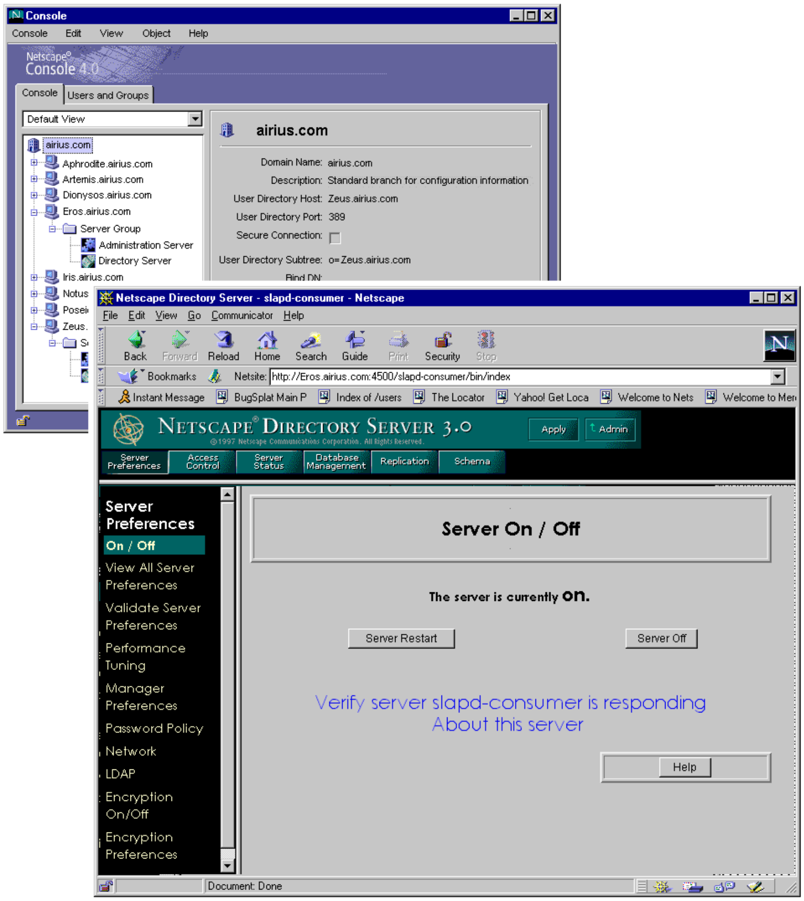
Adding a Pre-4.0 Server to the Tree

If you already have pre-4.0 Netscape servers installed in your enterprise, you can access it and continue to manage it though Netscape Console. This is useful when you want to keep and use your older pre-4.0 server until you're ready to put your 4.0 server into production.

The pre-4.0 server will not be completely integrated into the 4.0 environment: it will run in a browser as did before. For example, if you already have Netscape Messaging Server 3.0 installed, you can add it to the Navigation tree. But when you open the server, the 3.0 server appears in the browser. You manage it using the 3.0 Server Manager as you usually do.

If you want to fully integrate a pre-4.0 server into Netscape Console, you must migrate its configurations to a 4.0 installation.

Figure 3.3 When you open a pre-4.0 server, it opens in a browser.



airius.com

Domain Name: airius.com
Description: Standard branch for configuration information
User Directory Host: Zeus.airius.com
User Directory Port: 389
Secure Connection: □
User Directory Subtree: o=Zeus.airius.com
Bind DN:

Netsite: http://Eros.airius.com:4500/slapd-consumer/bin/index

Server On / Off

The server is currently on.

Verify server slapd-consumer is responding
About this server

Document: Done

To add a pre-4.0 server to the navigation tree:

1. In Netscape Console, from the Console menu, choose Add Pre-4.0 Server.
2. In the Add Pre-4.0 Server window, enter information for the server you want to add to the navigation tree, then click OK.
3. In the Server List window, all servers in the server root are, by default, checked. Deselect the servers you do not want to add to the navigation tree.
4. Click OK.

Migrating a Pre-4.0 Server to a 4.0 Server

When you migrate a pre-4.0 server, its configuration settings are copied to a newer 4.0 server installation. For example, if you're already using Netscape Messaging 3.0 Server, you can install the new Messaging 4.0 Server in a different server root. You could then migrate the 3.0 server settings to the 4.0 server for testing purposes.

Note The old and new servers can coexist on the same host system because they are installed in different server roots. **If you use the same port number for both servers, you cannot run both servers at the same time.** Before you start the 4.0 server, be sure the 3.0 server is turned off. Before you start the 3.0 server, be sure the 4.0 server is turned off.

Migrating the configuration settings takes less time than manually configuring the server. It also ensures that you maintain identical settings that worked for you in the older version. Once you're certain that the configuration settings work in the 4.0 server environment, you can safely remove your 3.0 server.

To migrate a pre-4.0 server to a 4.0 server:

1. Turn off the pre-4.0 server.
2. Run the Setup Program and install the 4.0 server. When prompted, specify a server root that's different from the 3.0 server root.
3. Start Netscape Console, and select the server group that contains the new 4.0 server. This group becomes the target group.

Note The Administration Server for the target group must be turned on, and you must have appropriate access privileges in order to proceed.

4. From the Object menu, choose Migrate Server Config.
5. In the Migrate Server Configuration window, enter the absolute path to the pre-4.0 server, then click OK.
6. In the Select Server for Migration window, select the pre-4.0 servers you want to migrate to 4.0 servers, then click Migrate.
7. In the Migrate Key and Certificate window:

- — If the pre-4.0 server uses SSL, provide the key password you used when you installed its SSL certificate, then click Migrate.
- — If the pre-4.0 server does not use SSL, click Cancel.

8. Restart the Administration Server for the target server group.

Cloning a Server

Cloning is useful when you want to replicate configuration settings on a number of servers of the same type. For example, an administrator has installed Netscape servers on each of ten hosts. The administrator configures the Administration Server on the first host. Then, instead of configuring each of the remaining nine Administration Servers individually, the administrator copies the configuration settings from the first one to each of the others.

When you clone a server, a predetermined number of attributes are copied from the reference server to the target server. The number and the actual attributes copied vary depending upon the server.

To clone server settings to another server:

1. In the navigation tree of Netscape Console, select the server that has the settings you want to replicate to other servers of the same type. This is the reference server.
2. From the Object menu, choose Clone Server.
3. In the Select Server window, choose the servers you cant to copy settings to (the target servers), then click OK.

Creating a New Server Instance

Once you have a server installed in a server root, you can create a new *instance*, or additional server of the same type, in the same server root. This is useful for testing purposes, and for when you need to share one server for two purposes.

For example, a company's Human Resources and Finance departments each need a Web server. Since each department has limited publishing requirements, one host can serve both departments' needs. The administrator installs one Web server, and the instance is for the Human Resources Department. The administrator then creates a new instance of the Web server for the Finance Department.

Note You cannot create a new instance of the Administration Server because each server root can have only one Administration Server.

To create a new instance:

1. In Netscape Console, select the server group that will contain the new server instance.
2. From the Object menu, select Create Instance Of.
3. In the Select Server window, choose the server from which you want to create the new instance, then click OK.

Removing a Server Instance

You can remove an instance of any server, other than the Directory and Administration Servers, from the navigation tree. This is not the same as uninstalling the server. When you uninstall the server, its program files are deleted from the host computer.

To remove a server instance:

1. In the Navigation Tree, select the server instance you want to remove.
2. From the Object menu, select Remove Server.

Uninstalling a Netscape Server

To remove a server from the navigation tree, you must uninstall the server.

To uninstall a server:

Unix. At the server root, enter`./uninstall.`

Windows NT. Use the Add/Remove Programs utility:

1. In the Start Menu, from the Settings menu, choose Control Panel.
2. In the Control Panel, choose Add/Remove Programs.
3. In the Add/Remove Program Properties window, choose Netscape Server Family 4.0, then click Remove.
4. In the Netscape Server Uninstall window, select the Netscape servers and components you want to uninstall, then click Uninstall.

Merging Configuration Data from Two Directory Servers

Once you've installed and deployed a number of Netscape servers, you might find it necessary to merge the existing configuration directory with a new one. For example, when you purchase a new server product that requires major changes in the existing configuration directory, you can stage or test the product against a pilot Directory Server. You can make adjustments to the pilot directory without impacting other servers or the existing directory in the enterprise. If you're using Netscape Console 4.1 or higher, once you're satisfied with the pilot directory, you can merge its configuration data into the configuration directory that's already deployed.

The Merge Configuration Directory utility copies all server instance entries (SIEs) in a Server Group, their configuration data, and all task entries in the pilot or *source* configuration directory. The utility then merges the data into the existing or *destination* configuration directory.

To merge configuration data from two Directory Servers:

1. In Netscape Console, in the navigation tree, right-click the Server Group containing the pilot (source) configuration directory, and then choose Merge Configuration.
2. In the Merge Configuration Directory Server Information window, enter information about the configuration directory you want to merge to:

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

   Destination LDAP Host. Enter the fully qualified host for the existing configuration directory you specified above.
   Example: `Zeus.Airius.com`

   Destination LDAP Port. Enter the port number for the existing configuration directory.

   Secure Connection. Select this option if the destination configuration directory uses the Secure Sockets Layer (SSL) protocol.

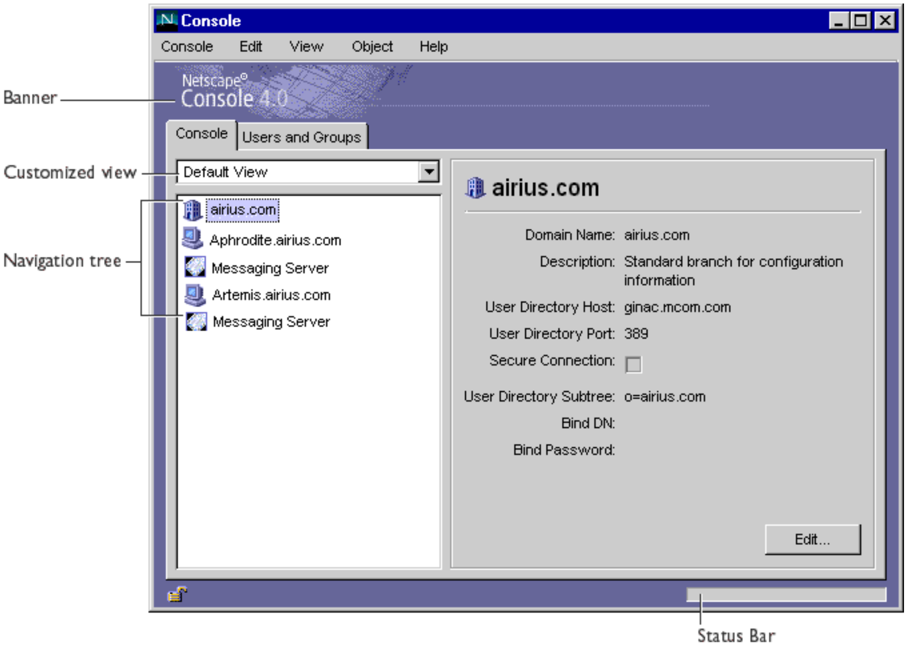
   Destination LDAP Bind DN. Enter the Distinguished Name for the user who has access to the destination configuration directory.

   Destination Bind Password. Enter the password for the user above.

After you merge the configuration directories, the affected SIE's will use the destination directory you specified. If you want the instances to switch back to the original configuration directory, you must manually modify the local configuration files. See "Changing the Configuration Directory Server" on page 140 for more information.

Customizing Your View of Netscape Console

To customize the appearance of Netscape Console, select or deselect items in the View menu. Select a menu item to display it, and deselect the item to hide it.



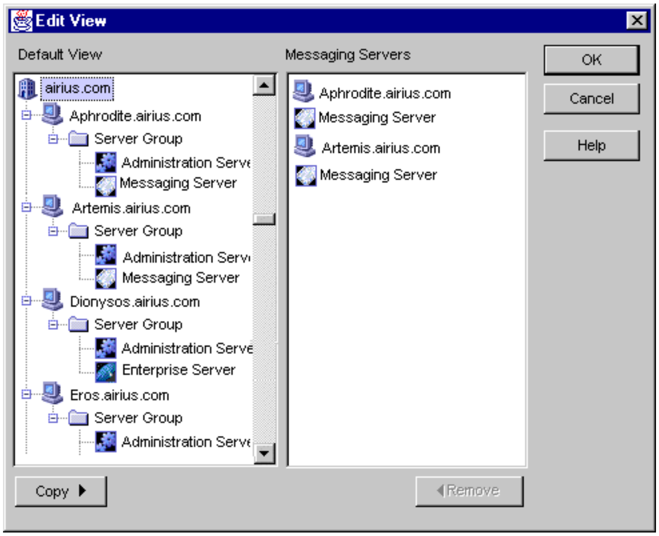
airius.com

Domain Name: airius.com
Description: Standard branch for configuration information
User Directory Host: ginac.mcom.com
User Directory Port: 389
Secure Connection: □
User Directory Subtree: o=airius.com
Bind DN:
Bind Password:

You can also customize the display of the navigation tree. This is useful when you want to show only those elements of your enterprise you access routinely, and hide elements you access infrequently.

To customize your view of the navigation tree:

1. From the View menu, choose New.
2. In the Edit View window, select an obejct from the tree on the left, then drag and drop it in the column on the right. In this example, the administrator has created a view named Messaging Servers, and built a view that includes only Messaging servers and their hosts.



3. Continue adding objects to the new view in this manner, and then click OK.

You can use multiple views to suit your needs. For example, an administrator frequently manages the company’s two Messaging Servers. He has a view named Messaging Servers that displays only on Messaging Servers and their respective hosts. When the administrator needs to see all the servers in the navigaction tree, he can go back to the Default view.

Display Preferences

By default, when you exit Netscape Console, it saves any display changes you’ve made in the session, such as

- window size or position

- showing or hiding the banner bar, status bar, or navigation tree
- fonts for menus, tables and other objects

You can store the display settings on the network or on your local disk to suit your individual needs.

Storing Display Preferences

To specify how and where you want settings stored:

1. In Netscape Console, from the Edit menu, choose Preferences.
2. Indicate how and where you want the display settings saved, then click OK.

   Where should preference settings be stored?

   - If you want to be able to use your settings no matter where you are when you log in to Netscape Console, choose **In Directory**. 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 Netscape Console you can use your preset display preferences.
   - If you want to be able to use different display settings depending upon the individual workstation you’re using, choose **On Local Disk** (stored in user’s home area). This is useful when you use one workstation at work and a dissimilar system, such as a laptop, at home. The settings that work for the workstation are stored and used on the workstation. The settings that work for the laptop are stored and used on the laptop.

   When should preference settings be saved?

   - If you want your most recent changes to be saved automatically, choose **On window close or console exit**. This is the default.
   - If you don’t want to save your changes automatically, choose **I will save them manually**.

   Save Now. Saves your most recent changes immediately.

   Reset. Reverts to all default display settings.

Setting Display Fonts

You can set preferences for fonts displayed in Netscape Console. You can save different sets of font preferences, or *profiles*, for multiple users of the same computer system.

To set display font preferences:

1. In Netscape Console, from the Edit menu, choose Preferences.
2. Enter font preferences, click OK.

   Profile. Enter a name for this set of preferences. If don't enter a name, a default name is provided for you.

   Save As. Saves the profile under the name you specify.

   Remove. Deletes a selected profile from the list.

   Change Font. displays a dialog box for setting your font preferences. In the Select Font dialog box, make font selections, then click OK.

Chapter

4

User and Group Administration

Netscape Console provides you access to a consolidated, networkwide repository for application data about user accounts, group lists, access privileges, and other security information. Use Netscape Console to create or locate and manage records for users and groups on any node in your enterprise.

This chapter contains the following sections:

- Interacting with the Directory Server
- End users can modify, but not create, a user entry.
- Modifying Existing Directory Entries
- Tracking User Licenses

Interacting with the Directory Server

When you use Netscape Console to create or modify users or groups, you make changes in the user directory, a subtree in the Directory Server. See “The Directory Server” on page 13 for a brief overview of Netscape Console architecture.

Using Distinguished Names

The User and Group interface of Netscape Console helps you create or modify Distinguished Names (DNs). Each user and group in your enterprise is represented in the Directory Server by a distinguished name (DN). A DN is a text string that contains identifying attributes. You use DNs whenever you make changes in the directory's users and groups database. For example, you need to specify DN information each time you

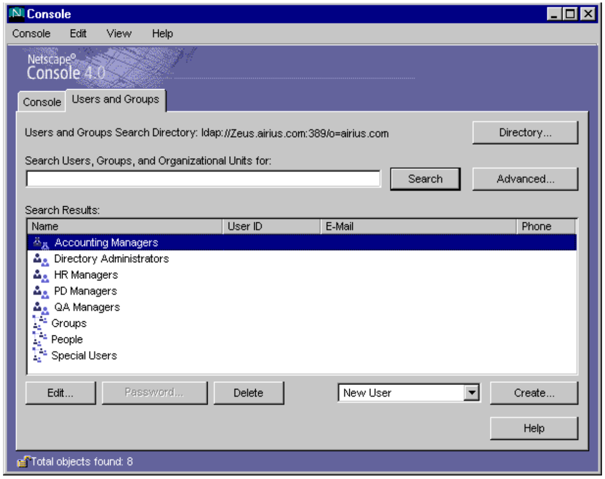
- create or modify directory entries
- set up access controls
- set up user accounts for applications such as mail or publishing.

See "Distinguished Name Attributes and Syntax" on page 149 for a brief summary of Distinguished Name syntax and frequently used attributes. For detailed information, see the *Administrator's Guide to Directory Server 4.0*.

Locating an Existing User or Group in the User Directory

The Users and Groups Search function works similarly to the basic Search function you find throughout Netscape Console. The search is performed against the default user directory. Any changes you make in the Users and Groups area of Netscape Console are made in the default user directory. You can manually change to a user directory other than the default. See "User Directory Settings" on page 142 for more information.

Figure 4.1 The Users and Groups area of Netscape Console.



Users and Groups Search Directory: ldap://Zeus.airius.com:389/o=airius.com

Search Users, Groups, and Organizational Units for:

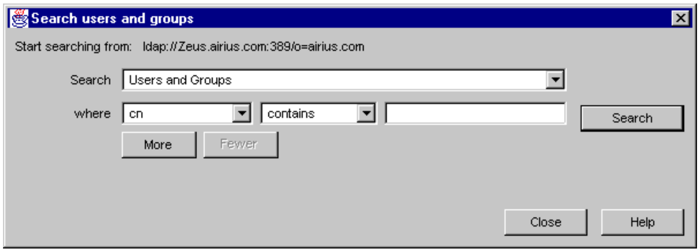
Search Results:

Name	User ID	E-Mail	Phone
Accounting Managers			
Directory Administrators			
HR Managers			
PD Managers			
QA Managers			
Groups			
People			
Special Users			

Total objects found: 8

To locate users or groups in the directory:

1. In Netscape Console, click Users and Groups.
2. In the Search field, enter a user or group name that can be found in the user directory.
3. (Optional) To specify more focused search criteria, click Advanced. In the Advanced Search dialog box, use the pull-down menus to first choose an attribute, then a search operator.



Start searching from: ldap://Zeus.airius.com:389/o=airius.com

Search Users and Groups

where cn contains

4. Click Search. Results are displayed in the list box.

Choosing a Different Search Directory

When you use the Users and Groups Search function, the URL for the default user directory appears. All searches are performed against this user directory. You can choose a user directory other than the default.

To change the search directory:

1. In Netscape Console, click Users and Groups.
2. Click 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 you want to use to connect to the user directory.

   User Directory Subtree. Use the form `o=airius.com` to indicate where to find the user directory.

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

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

4. Click OK.

End-User Access to the User Directory

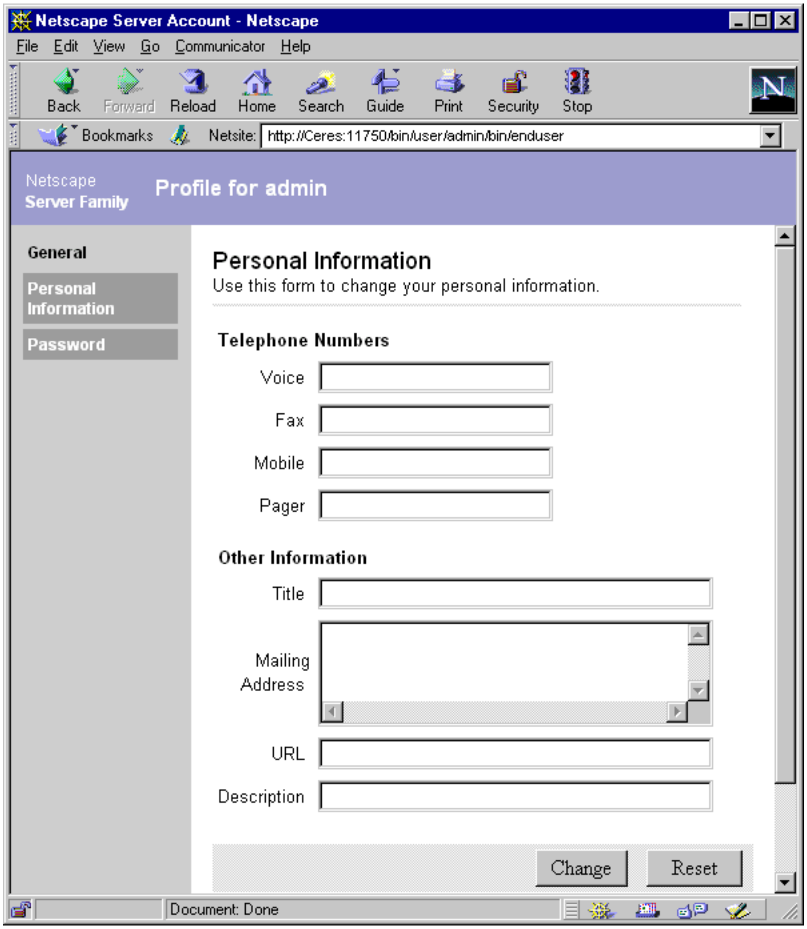
The end-user administration page is an HTML page designed to provide end users access to their own entries in the user directory. All users in the user directory are end users. For example, rank-and-file employees in your company might be given access to this page through a company phone book or directory. Using this page, shown in Figure 4.2, an employee can edit his own name, phone number, or other data that does not impact other directory entries. The changes made on this page are made in the default user directory.

To access the end-user administration page:

1. Open a browser, then enter the qualified host name and port number for the Administration Server you want to access.
   Example: `Venus.Airius.com:389`

2. In the Administration page, click Edit User Profile.

Figure 4.2 End users can modify, but not create, a user entry.



Netscape Server Account - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks
Netsite:
Netscape Server Family
General
Personal Information
Password
Change
Reset
Document: Done

http://Ceres:11750/bin/user/admin/bin/enduser

Profile for admin

Personal Information

Use this form to change your personal information.

Telephone Numbers

Voice

Fax

Mobile

Pager

Other Information

Title

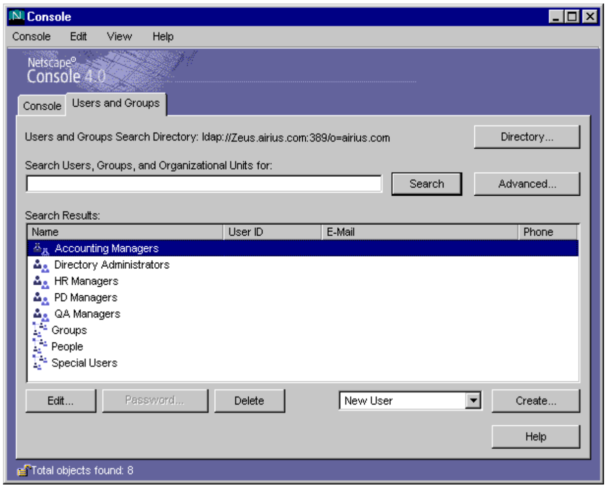
Mailing Address

URL

Description

Creating New Directory Entries

Go to the Users and Groups section of Netscape Console when you want to add or modify a user, group, or organizational unit. The User and Group graphical interface helps you create a DN entry in the directory.



Users and Groups Search Directory: ldap://Zeus.airius.com:389/o=airius.com

Search Results:

Name	User ID	E-Mail	Phone
Accounting Managers			
Directory Administrators			
HR Managers			
PD Managers			
QA Managers			
Groups			
People			
Special Users			

New User

Total objects found: 8

Note You can also use the command line to perform any of the directory operations described here. For detailed information, see the *Administrator's Guide to Directory Server 4.0*.

Organizational Units

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

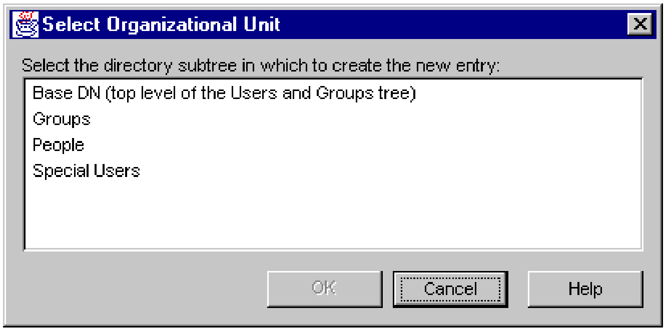
New organizational units are created using the organizational Unit object class. For example, if you create a new organization called Accounting within the organizational unit West Coast, and your Base DN is `o=Ace Industry, c=US,` then the new organization unit's DN is

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

Creating a new organizational unit

To create an organizational unit:

1. In Netscape Console, click Users and Groups.
2. Use the drop-down list to choose New Organizational Unit, then click Create.
3. In the Select Organizational Unit window, select the directory subtree (ou) to which the organizational unit will belong, then click OK.



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

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

   Name. Enter the name of the organizational unit.

   Description. Enter a description of the organizational unit that’s meaningful to you.

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

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

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

5. Click OK.

Groups

A group consists of all users who share a common attribute. For example, all users with DNs containing the attribute `ou=Sales` belong to the Sales group. Once you create a new group, you add users, or *members*, to it. You can use three types of groups in your directory: static, dynamic, and certificate groups.

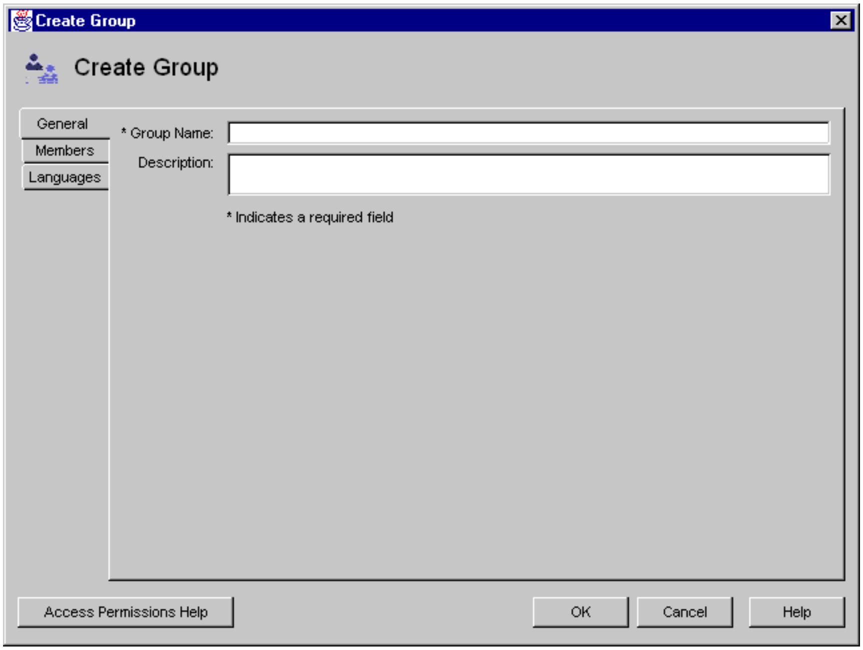
Creating a New Static Group

Create a *static* group by specifying the same group attribute in the DNs of any number of users. A static group doesn't change unless you add a user to it or delete a user from it. For example, a number of users have the attribute `department=marketing` in their DN. But none of those users are members of the Marketing group until you explicitly add each one to the group.

To create a static group in the directory:

1. In Netscape Console, click Users and Groups.
2. Use the drop-down list to choose New Group, then click Create.
3. In the Select Organizational Unit window, select the directory subtree (ou) to which the group will belong, then click OK.

4. In the Create Group dialog box, enter group information, then click Members.



Group Name. Enter a name for the group.

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

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

   To immediately add members to the group, click Members and then continue to the next step.

6. In the Members dialog box, click Add or Edit as appropriate, 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.

Creating a Dynamic Group

Create a dynamic group when you want users to be added automatically to a group based on their DN attributes. For example, you can create a group that automatically includes any DN that contains the attribute `department=marketing`. Whenever you apply a search filter for `deparment=marketing`, the search returns a group including all DNs containing the attribute `department=marketing`. The DNs are included automatically, without your having to add each individual to the group.

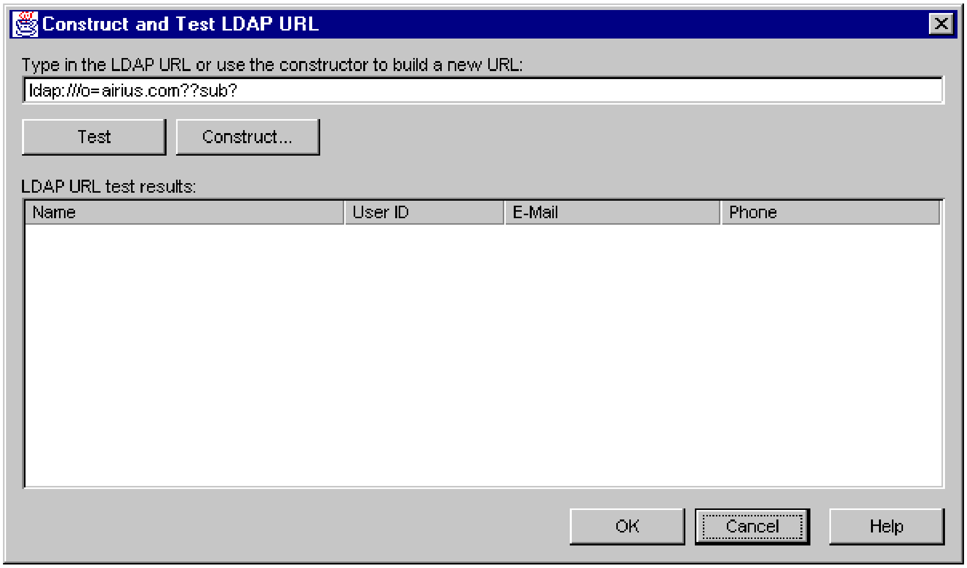
To create a dynamic group in the directory:

1. In Netscape Console, click Users and Groups.
2. Use the drop-down list to choose New Group, then click Create.
3. In the Select Organizational Unit window, select the directory subtree (ou) to which the group will belong, then click OK.
4. In the Create Group dialog box, enter general group information, then click Members.

   Group Name. Enter a name for the group.

   Description. (Optional) Enter a description to help you identify this group.
5. Click Dynamic Group, then click Add.

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



Construct and Test LDAP URL

Type in the LDAP URL or use the constructor to build a new URL: ldap:///o=airius.com??sub?

LDAP URL test results:

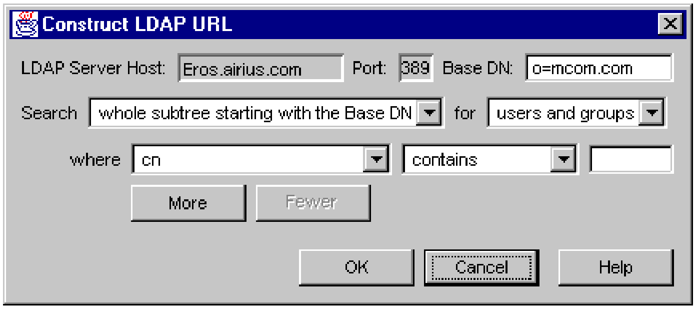
Name	User ID	E-Mail	Phone

Enter an LDAP URL and skip to step 8, or click Construct to build a new URL and continue to the next step.

The LDAP URL will take the form:

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

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



Construct LDAP URL

LDAP Server Host: Eros.airius.com Port: 389 Base DN: o=mcom.com

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

where cn contains

LDAP Server Host. Enter the fully qualified host name of the user directory you want to search. Example: <host>:<domain>

Port. Enter port number for the Directory Server that contains the specified user directory.

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

Search. Indicate the user directory subtree you want to search against.

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

where. Use the pull-down menus to first choose an attribute, then a search operator. Choices are described in the table below. In the last input field, enter a search string, then click Search.

More. Provides additional fields for specifying more attributes against which to search.

8. Click OK.
9. (Optional) In the Construct and Test LDAP URL dialog box, to see a list of users and groups included in the dynamic group, click Test.

   To accept the URL and add it to the list of dynamic group members, click OK.
10. Click Account, then select the accounts the group will use.
11. Click OK.

Creating a Certificate Group

Create a certificate group when you want to group all users who have a certificate containing a common attribute. For example, you can create a certificate for all users who share these attributes: ou=Sales, ou=West, ou=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 Western Sales group located in California. If the user's certificate does not contain these matching attributes, he is not recognized as a member of the group and does not receive the same access, privileges, or permissions as group members.

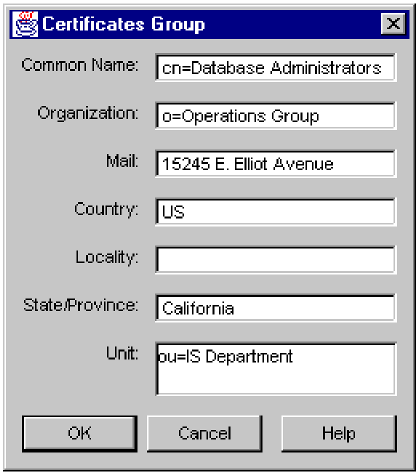
To create a certificate group in the directory:

1. In Netscape Console, click Users and Groups.
2. Use the drop-down list to choose New Group, then click Create.
3. In the Select Organizational Unit window, select the directory subtree (ou) to which the group will belong, then click OK.
4. In the Create Group dialog box, enter group information, then click Members.

   Group Name. Enter a name for the group.

   Description. (Optional) You can enter a description to help you identify this group.

5. Click Certificate Group, then click Add or Edit as appropriate.



Common Name: cn=Database Administrators

Organization: o=Operations Group

Mail: 15245 E. Elliot Avenue

Country: US

Locality:

State/Province: California

Unit: ou=IS Department

6. In the Certificate Group dialog box, provide the following information:

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

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

   Mail. Enter the street address of the groups' business.

Country. Enter the country code for the group's business.

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

State/Province. enter the state or province name for the group's business.

Unit. Enter the name of the unit within an organization that the group belongs to. Example: `ou=IS Department`

7. Click Account. Select the accounts the group will use.
8. Click OK.

Users

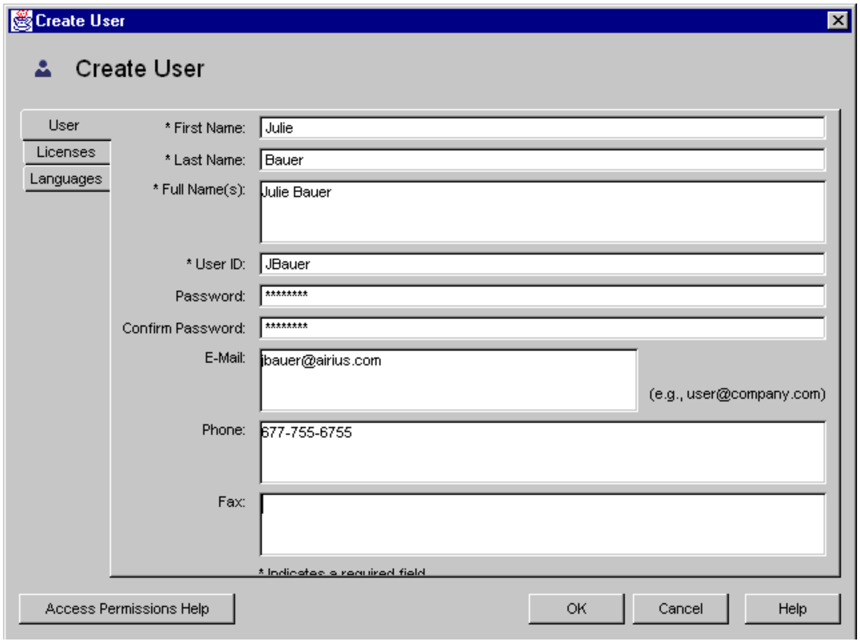
A user entry contains information about an individual person or object in the directory.

Creating a New User

To create a new user entry in the directory:

1. In Netscape Console, click Users and Groups.
2. Use the drop-down list to choose New User, then click Create.
3. In the Select Organizational Unit, select the directory subtree (ou) to which the user will belong, then click OK.

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



* First Name: Julie

* Last Name: Bauer

* Full Name(s): Julie Bauer

* User ID: JBauer

Password: ********

Confirm Password: ********

E-Mail: jbauer@airius.com (e.g., user@company.com)

Phone: 677-755-6755

Fax:

* Indicates a required field

First Name. Enter the user's full given name.

Last Name. Enter the user's full surname.

Full Name(s). This is equivalent to the common name (cn) in the directory and 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 userID must be unique from all other user ID's in the directory.

Password. (Optional) Enter the user's password.

Confirm Password. Enter the user's password again to confirm it.

E-Mail. (Optional) Enter the user's email address.

Phone. (Optional) Enter the user's telephone number.

Access Permissions Help. Provides information about setting access controls that apply to users and groups.

5. Click Licenses. Select the servers this user is licensed to use, then click OK.
6. Click Account. Select the accounts the user will use, than then click OK.
7. (Optional) Click Languages. Use the drop-down list to select the user's preferred language. Select (highlight) a language to see the Pronunciation field when appropriate.
8. (Optional) Enter language-related information:

   First Name. Enter the user's given name.

   Last Name. Enter the user's surname.

   Full Name(s). Enter the user's name as it should appear on official documents.

   Phone. Enter the user's telephone number.

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

9. Click OK.

The User's Preferred Language

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

To indicate a user's preferred language, follow the instructions in the section "Creating a New User" beginning on page 52.

Modifying Existing Directory Entries

Before you can modify user or group data, you must first use the User and Groups Search function to locate the user or group entry in the user directory. See “Locating an Existing User or Group in the User Directory” on page 44. Then you can select operations from the menu bar to change the entry. The operations you perform apply to all in the Search list.

Editing a User’s or Group’s Directory Entry

To edit a directory entry:

1. In the User and Group section of Netscape Console, use the Search function to locate the user or group.
2. Once the user or group name appears in the Search list, click it to select it, then click Edit.
3. Modify user or group information as necessary, then click OK.

Changing a User Password

To change a user password:

1. In the User and Group section of Netscape Console, use the Search function to locate and highlight the user.
2. Click Change Password.
3. Enter password as prompted, then click OK.

   New Password. Enter a password string. Alphanumeric characters, spaces, and punctuation marks are all acceptable.

   Confirmed Password. Enter the password again to confirm. The changes take effect immediately.

Removing a User, Group, or Organizational Unit from the Directory

Before you can remove an organizational unit, you must first remove all users or groups belonging to it.

To delete a user, group, or organizational unit from the directory:

1. In the User and Group section of Netscape Console, use the Search function to locate and highlight the user or group you want to delete.
2. Click Delete, and when prompted to confirm the deletion, click OK.

Tracking User Licenses

You can track which Netscape server products your users are licensed to use. This is useful when you need to report compliance with the software licensing agreement.

To view the number of users licensed to use Netscape products:

1. Go to Netscape Console.
2. From the File menu, choose License Tracking.
3. Select the servers you want to count licenses for, then click Refresh at the bottom of the dialog box.

The License Count column displays the number of licenses you have for each selected server.

Chapter

5

Using SSL

This chapter describes how to set up Netscape servers to support the Secure Sockets Layer (SSL) protocol. SSL has been universally accepted on the World Wide Web for authenticated, encrypted, and tamper-proof communication between clients and servers. Before reading this chapter, you should be familiar with the basic concepts described in "Introduction to Public-Key Cryptography" on page 185 and "Introduction to SSL" on page 219.

This chapter contains the following sections:

- The SSL Protocol
- Setting up SSL Encryption
- Obtaining and Installing a Certificate
- Activating SSL
- Managing Server Certificates
- Using Client Certificates

The SSL Protocol

The Secure Sockets Layer (SSL) protocol, which was originally developed by Netscape, 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.

Using External Encryption Devices

Public Key Cryptography Standard (PKCS) #11 defines the interface used for communication between SSL and PKCS #11 modules (also called cryptographic modules). Netscape support for PKCS #11 allows you to store server certificates on external devices such as smart cards, which can be desirable when server access must be restricted to just a few administrators.

A PKCS #11 module is a device, implemented in hardware or software, that provides cryptographic services such as encryption and decryption and (in some cases) storage of keys and certificates. Netscape provides a built-in software PKCS #11 module with its server and client products. Other kinds of PKCS #11 modules include the Netscape FORTEZZA module, used by the government, and the Litronic cryptographic module for smart card readers.

Netscape servers can use a variety of external PKCS #11 modules provided by different manufacturers. You must install the appropriate drivers provided by the manufacturer on the machine on which a Netscape server is running before you can use a given external module.

Slots and Tokens

A PKCS #11 module always has one or more slots, which may be implemented as physical hardware slots or conceptual slots implemented in software. Each slot in a PKCS #11 module can in turn contain a token, which is 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, and each slot can contain a token called a smart card.

An *internal token* is made up of a key-pair and a certificate database stored in a software file on a host computer. By default, the Netscape Administration Server provides a means to create an internal token with its PKCS #11 module. If you don't connect an external device such as PCMCIA card reader to your server and clients, then you can use the Netscape internal token for SSL authentication.

An *external token* is a key-pair and certificate database stored in an external device such as a Smart Card, FORTEZZA Card, or other Crypto Card. If you have an external device to your server, you can use external tokens for SSL authentication.

Setting Up an External PKCS #11 Module

To install the PKCS #11 Module:

1. Follow the instructions that came with your smart card reader or other hardware device to locate and install the appropriate drivers and connect the device.
2. In Netscape Console, open the server with which you want to use the external PKCS #11.
3. In the server's Console menu, choose Manage PKCS #11.
4. In the Manage PKCS #11 window, click Add.
5. In the Add PKCS #11 Module window, choose a file type. If the module is not contained in a JAR file, choose DLL, and then enter a name for the module. If the module is contained in a JAR file, choose JAR.
6. Enter the full path to the JAR file you obtained in step 1, and then click OK.

SSL Ciphers

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, company policies regarding acceptable encryption strength, and government restrictions on export of SSL-enabled software. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suites they will use to authenticate each other, to transmit certificates, and to establish session keys.

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.

Choosing SSL Ciphers

Decisions about which cipher suites a particular organization decides to enable depend on tradeoffs among the sensitivity of the data involved, the speed of the cipher, and the applicability of export rules.

Some organizations may want to disable the weaker ciphers to prevent SSL connections with weaker encryption. However, due to US government restrictions on products that support anything stronger than 40-bit encryption, disabling support for all 40-bit ciphers effectively restricts access to network browsers that are available only in the United States (unless the server involved has a special Global Server ID that permits the international client to “step up” to stronger encryption). For more information about US export restrictions, see Export Restrictions on International Sales.

To serve the largest possible range of users, it’s a good idea for administrators to enable as broad a range of SSL cipher suites as possible. That way, when a domestic client or server is dealing with another domestic server or client, respectively, it will negotiate the use of the strongest ciphers available. And when an domestic client or server is dealing with an international server or client, it will negotiate the use of those ciphers that are permitted under US export regulations.

However, since 40-bit ciphers can be broken relatively quickly, administrators whose user communities can use stronger ciphers without violating export restrictions should disable the 40-bit ciphers if they are concerned about access to data by eavesdroppers.

Note Netscape Console does not support all of the cipher suites supported by Netscape clients and servers. To ensure that Netscape Console can control an SSL-enabled server, the server must enable at least one of the following cipher suites for SSL 3.0:

- RC4 with 128-bit encryption and MD5 message authentication
- RC4 with 40-bit encryption and MD5 message authentication
- DES, which supports 56-bit encryption, with SHA-1 message authentication
- Triple DES, which supports 168-bit encryption, with SHA-1 message authentication
- No encryption, MD5 message authentication only

For detailed information on determining which cipher suites to use when setting up SSL, see Appendix E “Cipher Suites With RSA Key Exchange” on page 223 and “FORTEZZA Cipher Suites” on page 225**.**

Setting up SSL Encryption

All Netscape 4.0 servers support the SSL protocol and PKCS #11. Before you can use either one, you’ll need to do the following:

1. If you’re using an external token, install a PKCS #11 module.
2. Use the Certificate Setup Wizard to
   - create a key database and certificate database
   - create a public and private key-pair and install it into the key database
   - obtain and install a certificate

3. Enable SSL on your server.

The Certificate Setup Wizard automatically creates the key-pair and certificate databases for you. Once you've used the Certificate Setup Wizard to obtain and install your certificate, use Netscape Console to activate SSL on your server. The Certificate Setup Wizard is described fully in "Obtaining and Installing a Certificate" on page 67.

SSL Options

Before you start the Certificate Setup Wizard, determine whether you will use an internal or external token.

Using SSL with Internal Token

This option is the simplest to use. To set up SSL with an internal token, use the Certificate Setup Wizard. See "Obtaining and Installing a Certificate" on page 67 for more information. When prompted, specify the Internal token (cryptographic device).

SSL with External Token

This option requires the use of an additional hardware device as well as an external storage device such as a Crypto Card. The FORTEZZA cryptographic system, described in Appendix C, "FORTEZZA," uses an external token.

To set up SSL with an external token, first install the PKCS #11 module provided by the external device manufacturer. (See "Setting Up an External PKCS #11 Module" on page 63.) Then use the Certificate Setup Wizard as described in "Obtaining and Installing a Certificate", below. When prompted, specify the External token (cryptographic device).

SSL with Both Internal and External Tokens

Some servers in your enterprise may use only internal tokens, and some may use additional external tokens. Use this option if your server will communicate with both types of server. To set up SSL with both internal and external tokens, use the Certificate Setup Wizard *two times*. During the first use, when prompted, specify the Internal token. During the second use, when prompted, specify the External token.

Obtaining and Installing a Certificate

Use the Certificate Setup Wizard each time you need to request, renew, or install a certificate from a *Certificate Authority (CA)*. The first time you use the Certificate Setup Wizard, it creates and installs a *key-pair* and *certificate database* for you. For an introduction to each of these terms, see Appendix D, “A Certificate Identifies Someone or Something,” on page 193.

SSL Certificates

The Certificate Setup Wizard installs three types of certificates: a Server Certificate, a Server Certificate Chain, and a Trusted CA Certificate. You can install any number of certificates on a server. When setting up SSL for a directory server, you minimally need to install both a Server Certificate and a Trusted Certificate Authority certificate.

Server Certificate

This 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, first generate a request and send it to the CA. Then install the certificate. See “Generating a Server Certificate Request” on page 68 and “Installing the Certificate” on page 74.

Server Certificate Chain

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

Trusted CA Certificate

This is a single certificate automatically generated for you by your company’s internal certificate server, or by a CA known to you company. It’s used for client authentication. When you install a Trusted CA Certificate, by default it’s trusted.

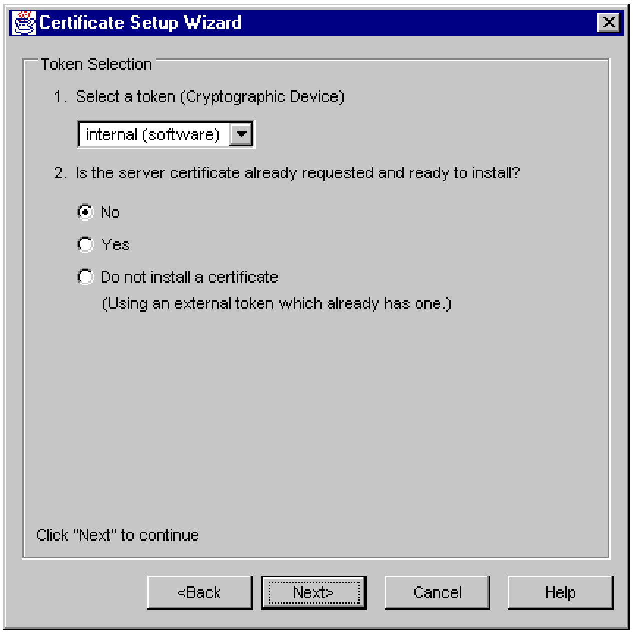
To obtain a Trusted CA Certificate, first go to the CA's website and copy the certificate information to your clipboard. Then use the Certificate Setup Wizard to install the certificate. See "Installing the Certificate" on page 74.

Generating a Server Certificate Request

Note After you've requested a certificate from a CA, it could take anywhere from two days to a number of weeks for the CA to send you a response and certificate.

To generate a certificate request and send it to a CA:

1. In Netscape Console, in the navigation tree, select the server you want to use SSL encryption with.
2. Click Open Server to open the server's console.
3. In Tasks, click Certificate Setup Wizard.
4. When prompted, provide information to request a certificate. Depending on certificates or tokens that may already be installed on the host system, you may see some or all of the following fields



Certificate Setup Wizard

Token Selection

1. Select a token (Cryptographic Device)

internal (software)

2. Is the server certificate already requested and ready to install?

● No

○ Yes

○ Do not install a certificate
(Using an external token which already has one.)

Click "Next" to continue

Select a token (Cryptographic Device). If your key will be stored in the local key database, choose internal (software). If your key will be stored in a SmartCard or other external device, choose the name of the external token.

Is the server certificate already requested and ready to install? If you've never submitted a request for this certificate, choose No. If your request has been processed and you already have a key for the certificate, or if you're installing a Trusted CA Certificate or certificate chain (which don't require a request), choose Yes. In this case, the wizard will skip to the steps described in "Installing the Certificate" on page 74.

Choose the third option if you're using FORTEZZA. (If you're using FORTEZZA, your certificte is already stored in an external device. Although you don't need to install a certificate, you do need to run the New Trust Database Setup program once.)

If a Trust Database doesn't already exist for this host, one is generated for you now. A Trust Database is a key-pair and certificate database installed on the local host. When you use an internal token, the Trust Database is the database into which you install the key and certificate. When you use an external token and the external device has insufficient storage capacity, the Trust Database stores your Certificate Revocation Lists (CRLs) and certificate chains.

5. Once a Trust Database is created, this message appears: "A Trust Database has successfully been created." Click Next.

6. Enter and confirm a password, then click Next.

   New Password. The password must contain at least eight characters, at least one of them numeric. This password helps secure access to the new key database you're creating.

   Confirm Password. Enter the same password again for verification.

7. Continue providing information as prompted, and click Next to go on.

   Is this a request for a new server certificate or to renew an existing server certificate? If you want to create a new certificate or replace an old one, choose New Certificate. If you already have an existing certificate, the Certificate Renewal option will take less time. If you have an existing certificate and want to replace or renew it, choose Certificate Renewal.

CA Email Address. Enter the CA administrator's email address to which your certificate request should be sent.

Show CA. Opens a browser and displays the Certificate Authorities available to you.

8. When prompted, enter the following information, and then click Next:

   Your name. Enter your full name.

   Telephone. Enter a telephone number where the CA can reach you if necessary.

   Server Host Name. Enter the fully qualified host name used in DNS lookups. Example: `<hostname>.<domain>`

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

   Organization. Enter the legal name of your company or institution. Most CAs require that you verify this information with legal documents such as a copy of a business license.

   Organizational Unit. (Optional) Enter a descriptive name for your organization within your company.

   Locality. (Optional) Enter your company's city name.

   State or Province. Enter the full name of your company's state or province (no abbreviations).

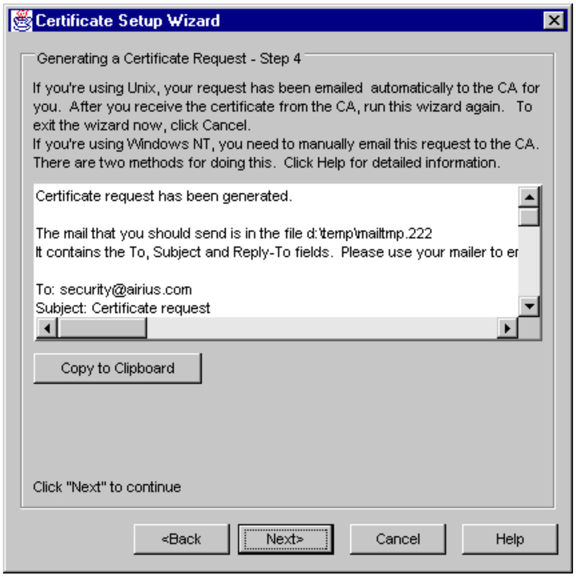
   Country. Enter the two-character abbreviation for your country's name (ISO format). The country code for the United States is US.

9. Enter the password for the token you selected earlier, and then click Next.

   Selected Token. Displays the token you'll be installing a certificate in.

   Trust Database Password. Enter the password you used when you set up the trust database.

The Certificate Setup Wizard generates a certificate request for your server. When you see this screen, you can send the certificate request to the CA. See “Sending the Server Certificate Request” for more information.



Generating a Certificate Request - Step 4

If you're using Unix, your request has been emailed automatically to the CA for you. After you receive the certificate from the CA, run this wizard again. To exit the wizard now, click Cancel.
If you're using Windows NT, you need to manually email this request to the CA. There are two methods for doing this. Click Help for detailed information.

```
Certificate request has been generated.

The mail that you should send is in the file d:\temp\mailtmp.222
It contains the To, Subject and Reply-To fields. Please use your mailer to er

To: security@airius.com
Subject: Certificate request
```

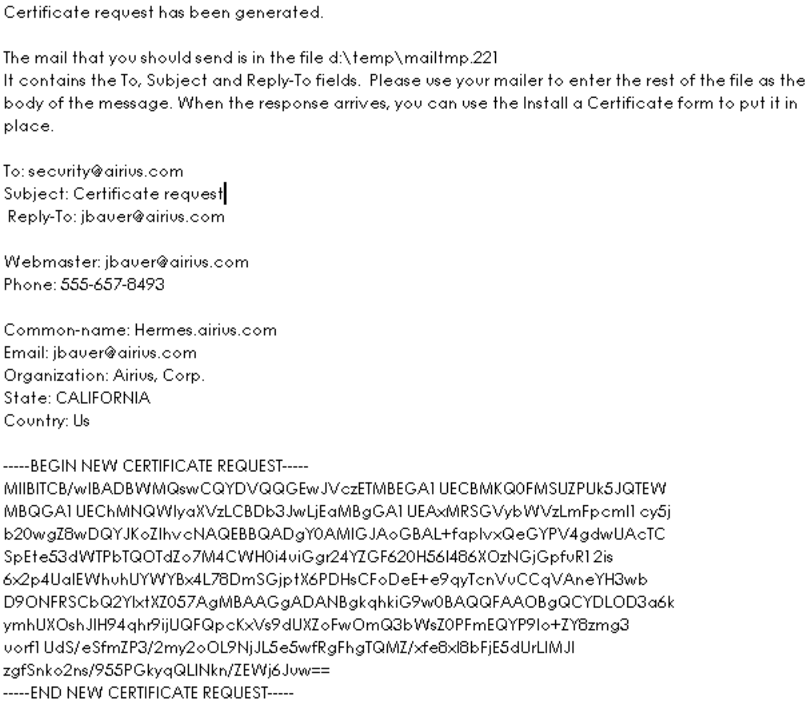
Click "Next" to continue

Sending the Server Certificate Request

If you're using Unix, the certificate request is sent for you automatically via email with sendmail.

If you're using Windows NT, the certificate information is automatically generated and saved into a temp file in the `\temp` directory. Follow these steps to send the certificate information to the CA:

1. Use your email program to create a new email message.
2. Manually open the temp file created for you in the`\temp` directory.

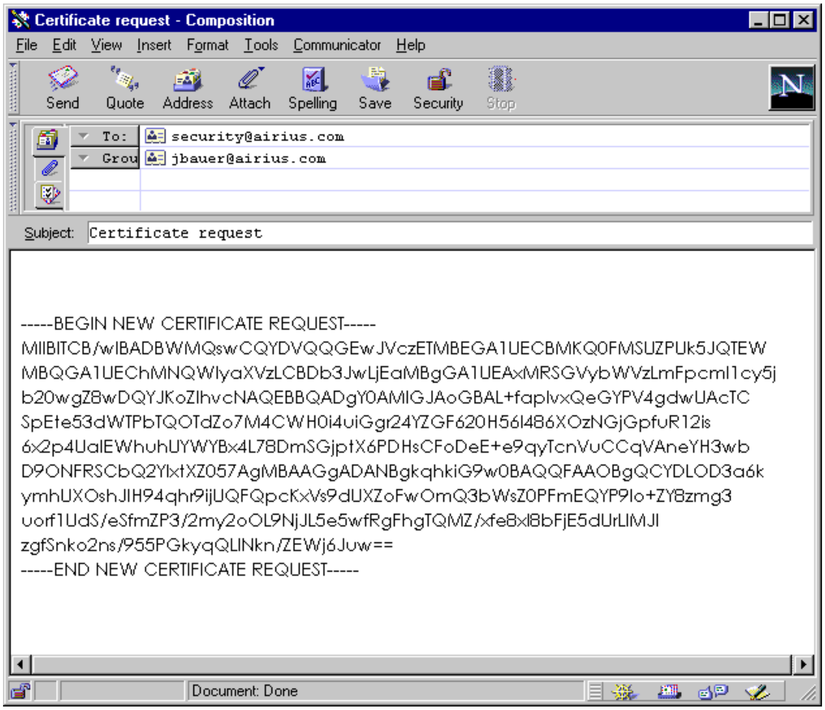


```
Certificate request has been generated.

The mail that you should send is in the file d:\temp\mailtmp.221
It contains the To, Subject and Reply-To fields.  Please use your mailer to enter the rest of the file as the
body of the message. When the response arrives, you can use the Install a Certificate form to put it in
place.

To: security@airius.com
Subject: Certificate request
 Reply-To: jbauer@airius.com

Webmaster: jbauer@airius.com
Phone: 555-657-8493

Common-name: Hermes.airius.com
Email: jbauer@airius.com
Organization: Airius, Corp.
State: CALIFORNIA
Country: Us

-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBITCB/wIBADBWMQswCQYDVQQGEwJVczETMBEGA1UECBMKQ0FMSUZPUk5JQTEW
MBQGA1UEChMNQWlyaXVzLCBDb3JwLjEaMBgGA1UEAxMRSGVybWVzLmFpcml1cy5j
b20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAL+faplvxQeGYPV4gdwUAcTC
SpEte53dWTPbTQOTdZo7M4CWH0i4viGgr24YZGF620H56l486XOzNGjGpfvR12is
6x2p4UalEWhvhUYWYBx4L78DmSGjptX6PDHsCFoDeE+e9qyTcnVvCCqVAneYH3wb
D9ONFRSCbQ2YlxtXZ057AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQCYDLOD3a6k
ymhUXOshJlH94qhr9ijUQFQpcKxVs9dUXZoFwOmQ3bWsZ0PFmEQYP9Io+ZY8zmg3
vorf1UdS/eSfmZP3/2my2oOL9NjJL5e5wfRgFhgTQMZ/xfe8xl8bFjE5dUrLIMJI
zgfSnko2ns/955PGkyqQLINkn/ZEWj6Jvw==
-----END NEW CERTIFICATE REQUEST-----
```

3. Copy the subject line from the temp file, then paste it into the subject line of the new message.
4. Copy the To address from the temp file, then paste it into the address field of the new message.

5. Copy the certificate information from the temp file, then paste it into the body of the new message.

   Your certificate request email should look like this:


Certificate request - Composition
File Edit View Insert Format Tools Communicator Help
Send
Quote
Address
Attach
Spelling
Save
Security
Stop
N

To: security@airius.com

Grou jbauer@airius.com

Subject: Certificate request

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBITCB/wIBADBWMQswCQYDVQQGEwJVczETMBEGA1UECBMKQ0FMSUZPUk5JQTEW
MBQGA1UEChMNQWlyaXVzLCBDb3JwLjEaMBgGA1UEAxMRSGVybWVzLmFpcml1cy5j
b20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAL+faplvxQeGYPV4gdwUAcTC
SpEte53dWTPbTQOTdZo7M4CWH0i4uiGgr24YZGF620H56I486XOzNGjGpfuR12is
6x2p4UalEWhuhUYWYBx4L78DmSGjptX6PDHsCFoDeE+e9qyTcnVuCCqVAneYH3wb
D9ONFRSCbQ2YlxtXZ057AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQCYDLOD3a6k
ymhUXOshJIH94qhr9ijUQFQpcKxVs9dUXZoFwOmQ3bWsZ0PFmEQYP9Io+ZY8zmg3
uorf1UdS/eSfmZP3/2my2oOL9NjJL5e5wfRgFhgTQMZ/xfe8xl8bFjE5dUrLIMJI
zgfSnko2ns/955PGkyqQLlNkn/ZEWj6Juw==
-----END NEW CERTIFICATE REQUEST-----
```

Document: Done

6. Send the email message to the CA.

   Once you've emailed your request, you must wait for the CA to respond with your certificate. Turnaround time for your request is highly variable and depends on the level of service provided by the CA. For example, if your CA is internal to your company, it may take only a day or two to respond to your request. If your selected CA is external, it could take several weeks to respond to your request.

7. After you've sent the email message to the CA

   - To exit the wizard and wait for the CA response, click Cancel.

   - When you receive a response from the CA, install the certificate as described in "Installing the Certificate" on page 74.

Backing Up Your Certificate Information

When the CA sends a response, you should back up all files in this directory: `<serverroot>/alias/`. This directory includes all the data for your Trust Database. You'll need the data when you install the certificate.

You should also save in a text file the certificate data you receive from the CA. If your system ever loses the certificate data, you can reinstall the certificate using your backup file.

Obtaining a Server Certificate Chain

To obtain a Server Certificate Chain:

1. Go to the CA's website and manually copy the certificate information to your clipboard.
2. Use the Certificate Setup Wizard to install the certificate. See "Installing the Certificate" on page 74.

When you install a certificate into an existing Server Certificate Chain, the new certificate is not automatically trusted. To trust a new certificate, you must change its trust option. See "Changing the CA Trust Option" on page 80.

Installing the Certificate

To install a certificate:

1. In Netscape Console, in the navigation tree, select the server you used when you generated the certificate request.
2. Click Open to open the server.

3. In Tasks, click Certificate Setup Wizard.

4. Start the Certificate Setup Wizard, and indicate that you're now ready to install the certificate. When prompted, provide the following information:

   Select a token (Cryptographic Device). Choose the same token you used when you generated the certificate request.

   Is the server certificate already requested and ready to install? Choose Yes.

   Certificate for: Indicate which type of certificate you're installing:

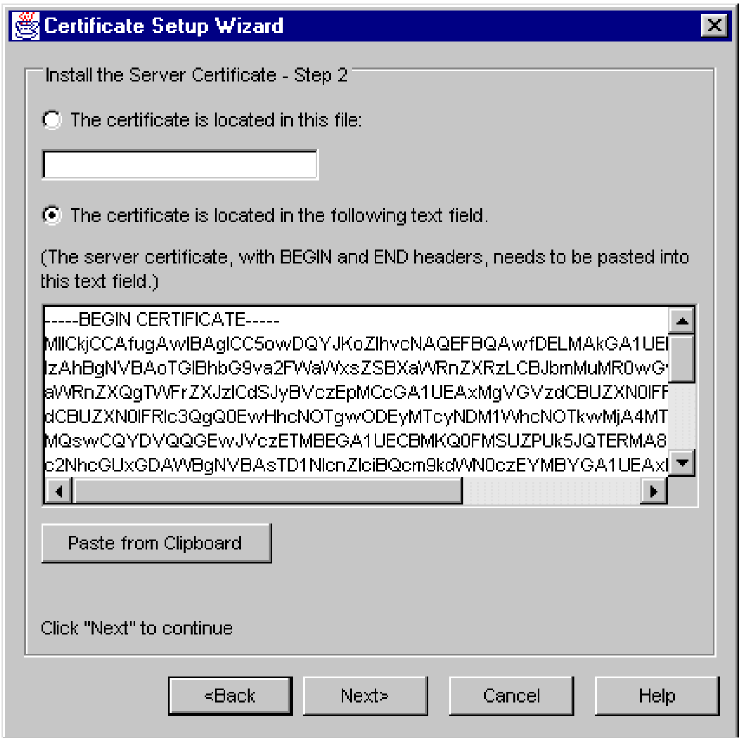
   — If you're using a single certificate associated only with your server, choose This Server.

   — If you're using a CA's certificate that includes a certificate chain, choose Server Certificate Chain.

   — If you're using a certificate that you want to accept as a trusted CA for client authentication, choose Trusted Certificate Authority.

   Password: Enter the password you used when you set up the Trust Database.

5. Provide the certificate information you received from the CA, then click Next.



Certificate Setup Wizard

Install the Server Certificate - Step 2

○ The certificate is located in this file:

● The certificate is located in the following text field.

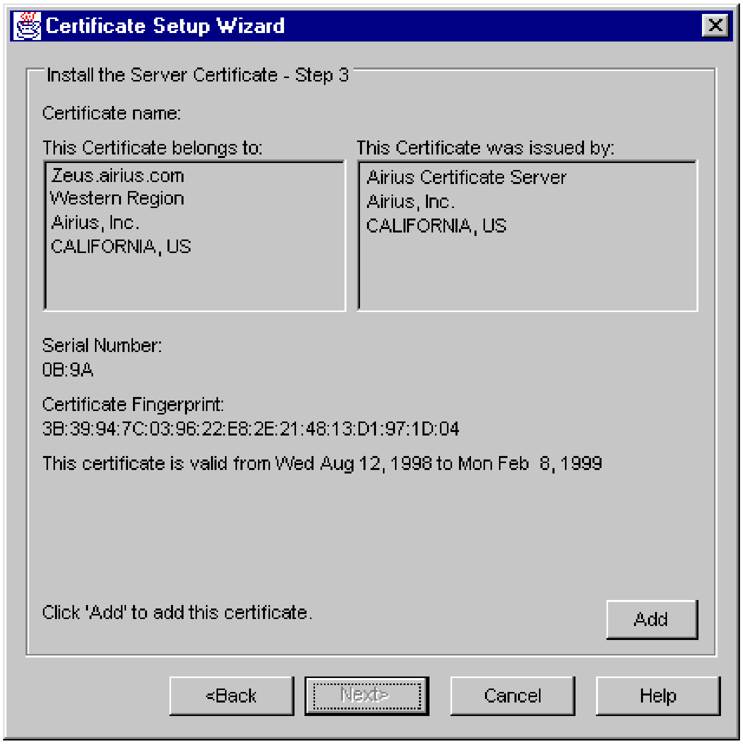
(The server certificate, with BEGIN and END headers, needs to be pasted into this text field.)

```
-----BEGIN CERTIFICATE-----
MIICkjCCAfugAwIBAgICC5owDQYJKoZIhvcNAQEFBQAwfDELMAkGA1UE
IzAhBgNVBAoTGlBhbG9va2FWaWxsZSBXaWRnZXRzLCBJbmMuMR0wG
aWRnZXQgTWFrZXJzICdSJyBVczEpMCcGA1UEAxMgVGVzdCBUZXN0IFF
dCBUZXN0IFRlc3QgQ0EwHhcNOTgwODEyMTcyNDM1WhcNOTkwMjA4MT
MQswCQYDVQQGEwJVczETMBEGA1UECBMKQ0FMSUZPUk5JQTERMA8
c2NhcGUxGDAWBgNVBAsTD1NlcnZlciBQcm9kdWN0czEYMBYGA1UEAx
```

The certificate is located in this file: Choose this option if you stored the certificate information in a text file. Enter the absolute path to the certificate.

The certificate is located in the following text field: Choose this option if you copied the certificate information directory from a CA's website to your system clipboard. Click Paste from Clipboard to insert the text in the input area.

6. Once the certificate is generated and you see this screen, click Add.



Certificate Setup Wizard

Install the Server Certificate - Step 3

Certificate name:

This Certificate belongs to:

Zeus.airius.com
Western Region
Airius, Inc.
CALIFORNIA, US

This Certificate was issued by:

Airius Certificate Server
Airius, Inc.
CALIFORNIA, US

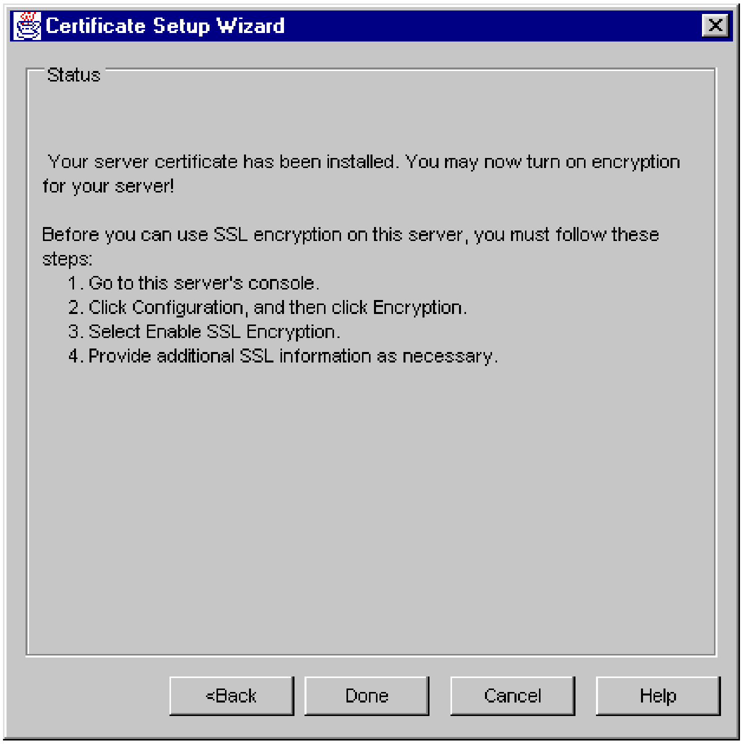
Serial Number:
0B:9A

Certificate Fingerprint:
3B:39:94:7C:03:96:22:E8:2E:21:48:13:D1:97:1D:04

This certificate is valid from Wed Aug 12, 1998 to Mon Feb 8, 1999

Click 'Add' to add this certificate.

7. Once your certificate is successfully installed, you'll see this screen. Click Done.



Your server certificate has been installed. You may now turn on encryption for your server!

Before you can use SSL encryption on this server, you must follow these steps:

1. Go to this server's console.
2. Click Configuration, and then click Encryption.
3. Select Enable SSL Encryption.
4. Provide additional SSL information as necessary.

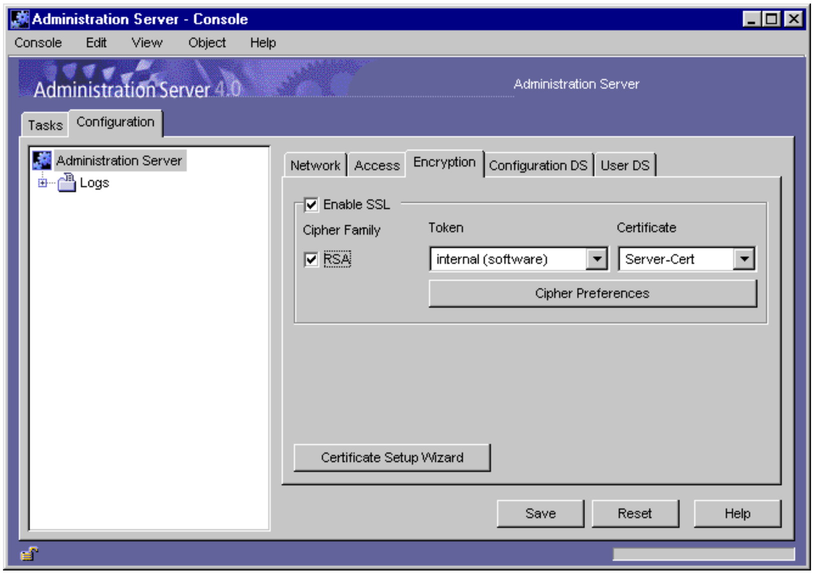
Activating SSL

Once you've obtained and installed a certificate, use Netscape Console to activate SSL. The following procedure uses the Administration Server as an example. Other Netscape server interfaces may vary slightly, although the basic concept is the same.

To activate SSL on a 4.x Netscape server:

1. In Netscape Console, in the navigation tree, select the server you want to use SSL encryption with.
2. Click Open, then click Configuration.

3. In the Configuration window, click Encryption.



☑ Enable SSL

Cipher Family: ☑ RSA

Token: internal (software)

Certificate: Server-Cert

4. Enter information as appropriate:

Enable SSL. Choose this option if you want to secure your enterprise 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.

Cipher Family. When you enable SSL Encryption, the cipher families available to you are listed here. Netscape currently supports two types of cipher families: RSA and FORTEZZA. The internal token supports only RSA. If you're using a FORTEZZA card, you'll see the FORTEZZA cipher family in this list. Select the cipher families you want to use.

Token to Use. Choose internal (software) if the key is stored in the local key database. All other choices available to you on this list are device-based. This means the key is stored on an external device such as a Smart Card.

Certificate to Use. Use the name of your server's certificate. If you're unsure which Certificate to use, view the Certificate Management dialog box for more information. To view the Certificate Management dialog box, from the Console menu, choose Certificate Management.

Cipher Preferences. A cipher is an encryption algorithm. This list displays the cipher preferences you've enabled.

5. Click OK or Save, as appropriate.

6. You won't be able to use Netscape Console to manage the now secured server until the changes saved, and you've restarted the server.Exit Netscape Console, then restart the server at the command line.

With changes in effect, you can start Netscape Console and log in as usual.

Managing Server Certificates

Once you've installed SSL certificates, you need to update information for them periodically. You can view, delete, or edit the trust settings of all the certificates installed on a server. This includes your own certificate and certificates from CAs.

Changing the CA Trust Option

You may need to reject a trusted CA temporarily. For example, you may be notified that a CA is experiencing technical difficulty that prevents certificate authentication. You can change the trust option to Reject until the CA notifies you that the problem has been resolved.

To change the key database trust option:

1. In Netscape Console, select a server and open it.

2. From the Console menu, choose Manage Certificate.

3. Select the certificate you want to update, then click Edit.

4. In the Certificate dialog box, make changes as necessary:

   Detail. Displays detailed information about the selected certificate including serial number, dates the certificate is valid, the Certificate Fingerprint, and whether or not the certificate is currently trusted.

   Delete. Deletes the selected certificate.

Trust/Reject. Allows the certificated to be trusted (or rejected).

5. Click OK.

Changing the Trust Database Password

It's a good practice to change your Trust Database password periodically. If your Administration Server is SSL enabled, your Trust Database password is required when starting the Administration Server. Changing your password periodically adds an extra level of server protection.

To change your Trust Database password:

1. In Netscape Console, select a server and open it.
2. From the Console menu, choose Change Key Password.
3. In the Change Key Password dialog box, enter password information:.

   Old password. Enter the password used to install the certificate.

   New Password. Enter a new password string.

   Confirm. Enter the password again to confirm it.

4. Click OK.

Managing Certificate Lists

The purpose of certificate revocation lists (CRLs) and compromised key lists (CKLs) is to make known any certificates and keys that either client or server users should no longer trust. If data in a certificate changes (for example, a user changes offices or leaves the organization) before the certificate expires, the certificate is revoked and its data appears in a CRL. If a key is tampered with or otherwise compromised, the key and its data appear in a CKL. Both CRLs and CKLs are produced and periodically updated by a CA.

Obtaining a CRL or CKL

To obtain a CRL or CKL from a Certificate Authority (CA):

1. Use a browser to go to the CA's website. Contact your CA administrator for the exact URL to use.
2. Follow the CA's instructions for downloading the CRL or CKL to a local directory.

Once you've saved the CRL file or CKL file to a local directory, you can add information from it to the Trust Database.

Adding a CRL or CKL to the Trust Database

To add CRL or CKL to the trust database:

1. In Netscape Console, select a server and open it.
2. From the Console menu, choose Certificate Revocation. All installed CRLs and CKLs are listed along with their expiration dates.
3. Click Add.
4. In the Add CRL/CKL dialog box, type the full path name to file, and indicate whether the path leads to a CRL or CKL.
5. Click OK. If the list already exists in the database, the list you specify here will replace the existing list.

Viewing, Adding, or Deleting CRLs or CKLs

To view, add, or delete CRLs or CKLs:

1. In Netscape Console, select a server and open it.
2. From the Console menu, choose Certificate Revocation. All installed CRLs and CKLS are listed along with their expiration dates.
3. Click a CRL or CKL to select it.
   - To add CRL or CKL to the trust database, click Add.

- To delete CRL or CKL from the trust database, click View. In the Certificate window, click Delete.

Using Client Certificates

Some Netscape Servers use client certificates to ensure authenticity when communicating with a client and to determine if a user has access to the server. Before you can use client certificates for authentication or access control, you must first fulfill these requirements:

- The Netscape server must have SSL turned on. See “Activating SSL” on page 78.
- The Administration Server must trust the CA who issued the certificate to the client. See “Managing Server Certificates” on page 80.
- The certificate must be mapped appropriately to the user directory. See “Editing the certmap.conf file” on page 84.

How Client Certificates Work

When the server gets a request from a client, it asks for the client’s certificate before proceeding. A Netscape client, such as Netscape Navigator or Netscape Communicator, sends the client certificate to the server. After checking that a client certificate chains up to a trusted CA, a Netscape server uses the `certmap.conf` file to look up the user’s entry in the directory and check the certificate presented for authentication against the certificate listed in the user’s entry. You edit one or more CA mappings in this file to determine how certificates issued by each CA should look up user entries. Specifically, `certmap.conf` provides three kinds of information for each CA:

4. It maps the distinguished name (DN) in the certificate to a branch point in the LDAP directory.
5. It tells the server what values to use from the DN in the certificate (such as the user’s name, email address, and so on) for the purpose of searching the directory.

6. It specifies whether or not the server goes through an additional verification process. If the `certmap.conf` file is configured to support single sign-on, this process involves matching the certificate presented for authentication with the certificate stored in the user's LDAP directory entry. This step allows you to revoke a certificate by removing it from the user's entry in the directory. This prevents authentication even if the certificate is otherwise valid.

If it finds more than one matching entry, the server can verify the client's certificate by comparing it with certificates for the matching entries in the LDAP directory. If the client certificate doesn't match any certificates in the matching entries or if the matching entries don't contain certificates, the certificate mapping (and thus client authentication) fails.

After the server finds a matching entry and certificate in the LDAP directory, it can use that information to 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 ACLs to determine what resources the user is authorized to access.

Editing the certmap.conf file

The certificate mapping file is located at `<server_root>/shared/config/certmap.conf.` The file defines

- where in the LDAP tree the server should begin its search.
- what certificate attributes the server should use as search criteria when searching for the entry in the LDAP directory
- whether or not the server goes through an additional verification process

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

```
certmap name issuerDN

name:property [value]
```

The first line specifies a name for the entry and the DN of the issuer of the client certificate. The name is arbitrary; you can define it to be whatever you want.

Note The issuerDN must exactly match the issuer DN of the CA that issued the client certificate. For example, the following two issuer DN lines differ only in the number of spaces separating the AVAs, but the server treats these two entries as different:

```
certmap moz ou=Mozilla Certificate
Authority,o=Netscape,c=US
```

```
certmap moz ou=Mozilla Certificate Authority, o=Netscape,
c=US
```

The second and subsequent lines in the named mapping match properties with values. The certmap.conf file has six default properties:

- DNComps is a list of comma-separated DN attribute tags used to determine where in the LDAP directory the server should start searching for directory entries that match the user's information (that is, the owner of the client certificate). The server gathers values for these tags from the client certificate and uses the values to form an LDAP DN, which then determines where the server starts its search in the LDAP directory. For example, if you set DNComps to use the o and c DN attribute tags, the server starts the search from the o=org, c=country entry in the LDAP directory, where org and country are replaced with values from the DN in the certificate.
    - 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.
    - If the DNComps entry is present but has no value, the server searches the entire LDAP tree for entries matching the filter.

The following component tags are supported for DNComps: cn, ou, o, c, l, st, e, and mail. Case is ignored. You can use e or mail, but not both.

- FilterComps is a list of comma-separated DN attribute tags used to create a filter by gathering information from the user's DN in the client certificate. The server uses the values for these tags to form the search criteria for matching entries in the LDAP directory. If the server finds one or more entries in the LDAP directory that match the user's information gathered from the certificate, the search is successful and the server optionally performs a verification. For example, if FilterComps is set to use the e and uid attribute tags (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 LDAP database. The following component tags are supported for FilterComps: cn, ou, o, c, l, st, e, and mail. Case is ignored. You can use e or mail, but not both.

- verifycert tells the server whether it should compare the client's certificate with the certificate found in the user's LDAP entry. It takes two values: on and off. Netscape recommends that you set this to on for a complete single sign-on solution. This ensures that the server will not authenticate the client unless the certificate presented exactly matches the certificate stored in the directory. To revoke a user's certificate, you can just remove it from the user's LDAP entry.

- CmapLdapAttr is the name of the entry attribute in the LDAP 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 (see the Directory Server *Administrator's Guide* for details). If the CmapLdapAttr property exists in the certmap.conf file, the server searches the entire LDAP directory for an entry whose attribute (named with this property) matches the subject's full DN (taken from the certificate). If the search doesn't yield any entries, the server retries the search using the DNComps and FilterComps mappings. The search will take place more quickly if CmapLdapAttr is an indexed LDAP attribute.

  This approach to matching a certificate to an LDAP entry is useful when it's difficult to match entries using DNComps and FilterComps.

- Library is 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 perform the actual mapping on the basis of information in the certmap.conf file. (This is typically not necessary unless you have very specialized mapping requirements.)

- 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 standard functions that perform the actual mapping on the basis of information in the certmap.conf file. (This is typically not necessary unless you have very specialized mapping requirements.)

You can use the client certificate API to create your own properties. For information on programming and using the client certificate API, see your server's documentation or release notes.

Once you have a custom mapping, you reference the mapping as follows:

```
<name>:library <path_to_shared_library>

<name>:InitFn <name_of_init_function>
```

For example:

```
certmap default1 o=Netscape Communications, c=US

default1:library /usr/netscape/suitespot/userdb/plugin.so

default1:InitFn plugin_init_fn

default1:DNComps  ou o c

default1:FilterComps l

default1:verifycert on
```

Example mappings

The `certmap.conf` file should have at least one entry. The following examples illustrate two different ways you can use the `certmap.conf` file.

Configuration Example #1

Here is a simple `certmap.conf` file with only one default mapping:

```
certmap default default

default:DNComps ou, o, c

default:FilterComps e, uid

default:verifycert on
```

Using this example, the server starts its search at the LDAP branch point containing the entry `ou=orgunit, o=org, c=country`, where the italics represent values from the subject's DN in the client certificate.

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

Configuration Example #2

Here is another sample file:

```
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
```

This file has two mappings: a default one and another for MyCA. When the server gets a certificate from anyone other than MyCA, the server uses the default mapping, which starts at the top of the LDAP tree and searches for an entry matching the client's email address and user ID. If the certificate is from MyCA, the server starts its search at the LDAP branch containing the organizational unit and searches for matching email addresses. Also note that if the certificate is from MyCA, the server verifies the certificate; other certificates are not verified.

Note The issuer DN (that is, the CA's information) in the certificate must be identical to the issuer DN listed in the first line of the mapping. Even an extra space after a comma will cause a mismatch.

Configuration Example #3

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

```
certmap myco ou=My Company Inc, o=myco, c=US

myco:CmapLdapAttr certSubjectDN
```

```
myco:DNComps  o, c

myco:FilterComps mail, uid

myco:verifycert on
```

If the client certificate subject is

`uid=Henry Jones Junior, o=Ark Inc, c=US`

the server first searches for entries that have

`certSubjectDN=uid=Henry Jones Junior, o=Ark Inc, c=US`

If one or more matching entries are found, the server proceeds to verify the entries. If no matching entries are found, the server will use DNComps and FilterComps to search for matching entries. In this example, the server would search for `uid=Henry Jones Junior` in all entries under

`o=Ark Inc, c=US.`

Chapter

6

Delegating Server Administration

Through the use of administrative privileges and Access Control Information (ACIs) you can delegate specific server management tasks to selected individuals as you deem appropriate.

Note Each Netscape server has its own specialized functions, and each server has its own special types of ACIs. For detailed information about ACIs for a particular Netscape server, see the server's *Administrator's Guide*.

This chapter contains the following sections:

- Overview of Delegated Administration
- Access to Network Resources
- Access to Server Tasks

Overview of Delegated Administration

When a user logs into Netscape Console, the Administration Server authenticates the user against the Directory Server. During authentication, the Administration Server evaluates the user's administrative privileges and any Access Control Information (ACIs) pertaining to the user. When authentication is completed, Netscape Console displays only the resources and server tasks the user is allowed to access.

Delegating server administration is a two-step process. First, you provide specific users and groups with *administrative privileges*, or access, to various *resources*, such as host systems and servers in your enterprise. Once you've given administrative privileges to an individual, you can restrict the scope of the administrator's network or server responsibilities.

Network Resources and Administrative Privileges

All network resources registered in the same configuration directory form a Netscape *topology*. The entire navigation tree in Netscape Console represents a Netscape topology. An *administration domain* is a collection of host systems and servers that share the same user directory. A *server group* consists of all servers managed by the same Administration Server. *Servers* are the products that provide specific services such as directory, messaging, and publishing.

Netscape Console uses four levels of administration privileges to determine whether individuals are authorized to access network resources. Three levels of administration privileges correspond to entries in the user directory: Configuration Administrator, Domain Administrator, and Server Administrator. A fourth level, the Administration Server Administrator, has privileges only to the local Administration Server. A comparison of administrators and their corresponding privileges is summarized in Table 6.1.

The Configuration Administrator and the Administration Server Administrator are automatically created when you install Netscape Console. You manually create the Domain Administrator after you create an administration domain (See "Creating an Administration Domain" on page 29).

For more information on the Server Administrator, see the documentation that comes with your server.

Table 6.1 Summary of Administrative Privileges

Administrator	Primary Purpose	Description	Scope of Administrative Privileges
Configuration Administrator	To manage servers and configuration directory data in the entire Netscape topology.	When a configuration directory is first installed, the Configuration Administrators group and the `Configuration Administrator` user ID are both automatically created in configuration directory. Initially manages Administrative Domain configuration until the Domain Administrators group and its members are in place.	Unrestricted access to all resources in the Netscape topology. This is the only administrator who can assign Domain Administrators; can also provide server access to other administrators.
Domain Administrator	To manage servers and user data in an administrative domain.	Configuration Administrator must manually create a Domain, then assign a Domain Administrator to it. Domain Administrator can set access permissions for a server group, or for an individual server.	Restricted access to all servers and user data in a domain; can provide server access permissions to other administrators.
Server Administrator	To perform server management tasks.	Configuration or Domain Administrator must provide this user access to a server. Once a user has server access permissions, he is a Server Administrator and can provide server access permissions to others.	Restricted access to tasks for a particular server, depending upon task ACIs.
Administration Server Administrator	To start or stop a server even when there is no Directory Server connection.	When an Administration Server is installed, this administrator's entry is automatically created locally. (This administrator is not a user in the user directory.)	Restricted server tasks (typically only Restart Server and Stop Server) for all servers in a local server group.

Configuration Administrator

During installation, you're asked to specify a username and password for the Configuration Administrator. The Configuration Administrator is authorized to access and modify the Configuration Directory of your LDAP server. Netscape Console creates the Configuration Administrator as an entry in the LDAP user directory under: `ou=Administrators, out=Mission Control, ou=<domain>, o=NetscapeRoot`.

Normally, when you log in to Netscape Console as the Configuration Administrator, the username and password you enter are authenticated against the LDAP entry. But if the Directory Server cannot be accessed or the user LDAP entry cannot be found, Netscape Console authenticates the username and password against the Administration Server Administrator's credentials.

Administration Server Administrator

The Administration Server Administrator can execute limited `.cgi` programs such as starting, stopping, or restarting servers in the local Server Group. It was designed to provide a means for you to log in the Netscape Console when the Directory Server is not running.

During installation, Netscape Console uses the same username and password you specified for the Configuration Administrator to automatically create the Administration Server Administrator username and password.

The Administration Server Administrator does not have an LDAP entry; it exists only as an entity named in a local configuration file stored at `<server_root>/admin-serv/config/admpw`. The user id and password stored in this file are used for authentication when the Directory Server cannot be reached. This is what makes it possible for you to access an Administration Server and perform limited server administration even when the Directory Server is not running.

Changing Administrator Usernames and Passwords

Keep in mind that the Configuration Administrator and Administration Server Administator are two separate entities even though they are created at the same time during installation. If you change the username or password for one, Netscape Console does not automatically make the same changes for the other.

To change the username or password for the Configuration Administrator:

1. In Netscape Console, click Users and Groups.
2. In the Users and Groups window, click Directory.
3. .In the Change Directory Window, enter a new Bind DN or Bind Password, then click OK.

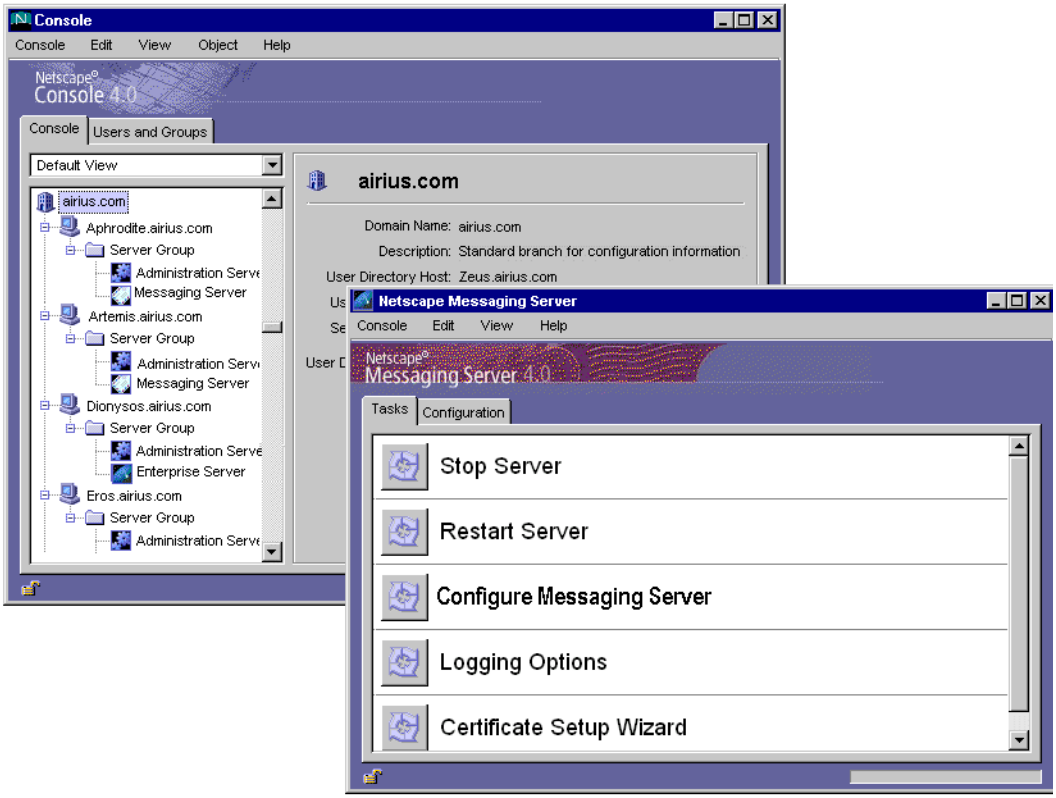
To change the username or password for the Local Administrator:

1. In the Netscape Console navigation tree, locate and select the Administration Server you want to reconfigure. Click Open to open the Administration Server window.
2. In the Administration Server window, click Configuration.
3. In the Configuration tab, click Access.
4. In the Access tab, enter a new Username or Password.
5. Restart the Administration Server.

Examples of Delegated Administration

Jane is an administrator who troubleshoots network problems for end users. She needs to be able to access any server in any domain, and frequently modifies many types of user account information. She has a wide range of access permissions. When Jane logs into Netscape Console, she has a relatively unrestricted view of servers and tasks.

Figure 6.1 A member of the Administrator's group has an unrestricted view of network resources and server tasks.



airius.com

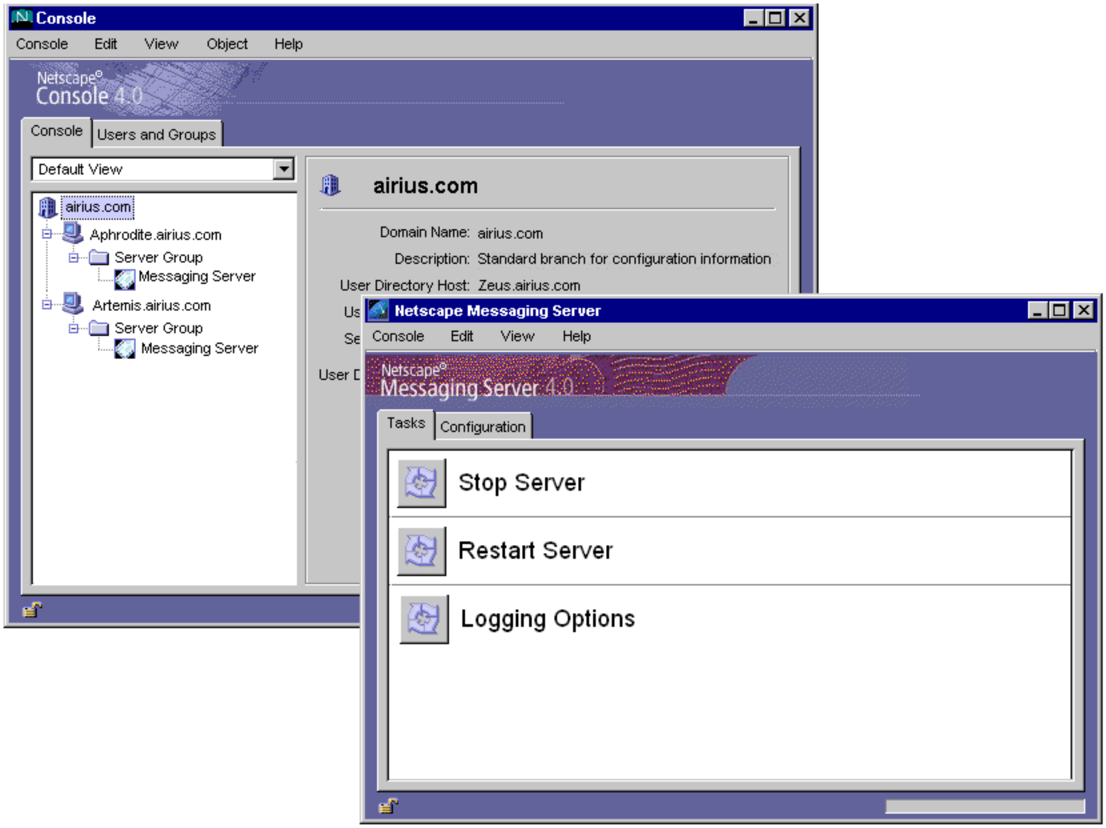
Domain Name: airius.com

Description: Standard branch for configuration information

User Directory Host: Zeus.airius.com

John is also an administrator, but his job is focused on managing mail servers in the network. John's access permissions are more limited than Jane's. John is only allowed to access mail servers and can only modify user information related to mail accounts. When John logs into Netscape Console, he sees only the servers and tasks he needs to see in order to do his job

Figure 6.2 A member of the Messaging Administrators group sees only the servers and tasks assigned to him.



Domain Name: airius.com
Description: Standard branch for configuration information
User Directory Host: Zeus.airius.com

Access to Network Resources

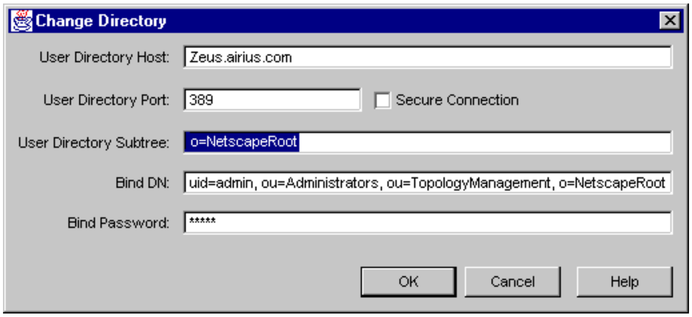
You provide access to network resources by adding users to administrators groups or by setting access permissions for a particular server.

Adding Users to the Configuration Administrators Group

Note The Configuration Administrators group is automatically created when the configuration directory is installed. Only members of the Configuration Administrators group can add more users to the group. Members of the Configuration Administrators group have unrestricted access permissions.

To add users to the Configuration Administrators group:

1. In Netscape Console, click Users and Groups, then click Directory.
2. In the Change Directory window, indicate the location of the user directory that contains the Configuration Administrators group, then click OK.



User Directory Host: Zeus.airius.com

User Directory Port: 389 □ Secure Connection

User Directory Subtree: o=NetscapeRoot

Bind DN: uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

Bind Password: *****

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.

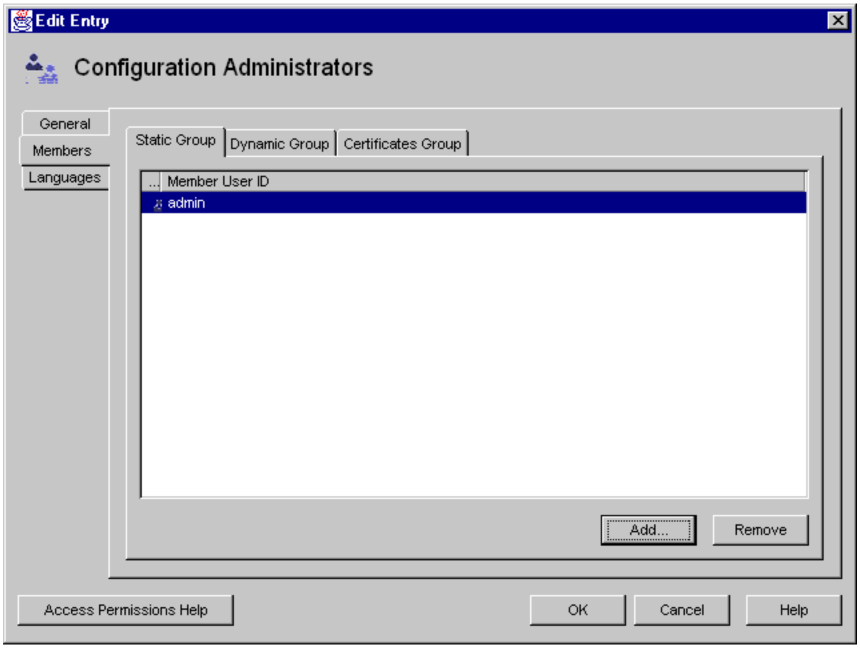
User Directory Subtree. Enter `o=NetscapeRoot` to indicate where to find the Configuration Administrators group.

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

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

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

4. In the Edit Group window, click Members.



5. Click Add.

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

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

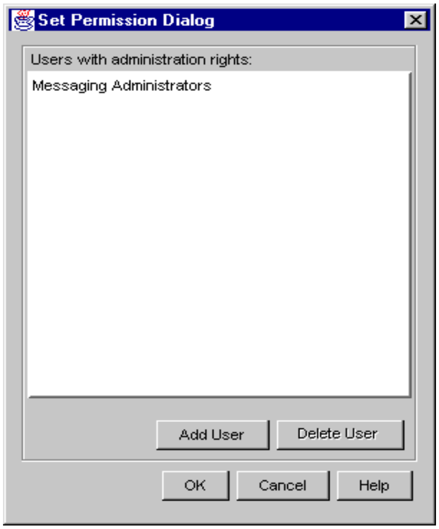
Setting Access Permission for an Individual Server

Users who have access permissions to a particular server can provide the same access to additional users. By default, the Configuration Administrator has the appropriate access permissions; Domain-level adminstrators and server administrations who've been given access permissions for an individual server can also provide the same access to other users.

To set access permissions for an individual server:

1. In Netscape Console, select the server you want to allow or deny access to.
2. From the Object menu, choose Set Access Permissions, and a list appears. The list contains the names users and groups who currently have access permissions for the selected object.

   By default, the Configuration Administrators group has unrestricted access to all servers, even though its name does not display on this list.



Messaging Administrators

3. To deny access permission to a user or group in the list, select the user or group name, then click Delete User. Skip the rest of this procedure.

To allow access permission to additional users or groups, click Add User.

4. Use the Search dialog box as usual to locate the user or group you want to allow or deny access permissions to, then click OK.

5. In the Set Access Permissions dialog box, be sure that the user or group is added to the list, then click OK.

Access to Server Tasks

You provide access to server tasks by creating Access Control Information (ACI) rules. ACI rules determine who has permission to perform specific server tasks such as starting, stopping, or configuring a server. The ACI Editor is a graphical interface that helps you create Access Control Information or *rules*. (See the illustration in “Setting Access Permissions for a Server Task” on page 103.)

Note Each Netscape 4.0 server may have its own ACI extensions and different uses for the ACI Editor. For detailed information about a particular server’s ACI options, see the *Administrator’s Guide* for that server.

What’s in an ACI

Each entry in the user directory maintained by a Directory Server can include one or more ACI *attributes*. Attributes contain access control information for the entry. The access control information is composed of three parts: a target, permissions, and bind rules.

Target

The target specifies the object, object attributes, or group of objects and attributes you’re controlling access to.

Permissions

The permission specifically outlines what rights you are either allowing or denying. Read, write, and execute are typical access permissions specified in ACIs. See Table 6.1 on page 93 for a brief summary of access permissions.

Bind Rules

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

- the user or group allowed or denied access permissions
- host computers from which users are allowed or denied access
- an interval of time during which the user's access is allowed or denied
- the type of permissions to be granted or denied to users or groups

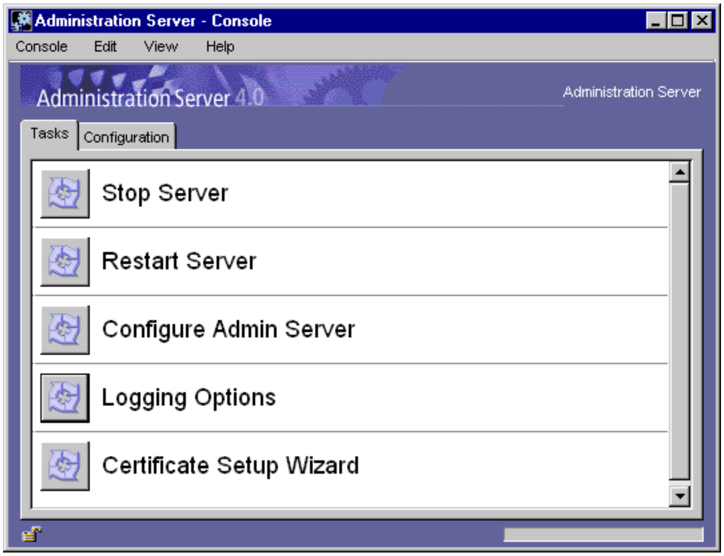
ACI attributes are stored in the Directory Server entry for the targeted resource. The following example illustrates the use of two ACIs in the same directory entry. The first ACI allows all members of the Directory Administrators group unrestricted access to the Directory Server. The second ACI denies access to the Directory Administrators group from 1:00 a.m. to 3:00 a.m. (0100 to 0300) on Sunday, Tuesday, and Friday:

```
dn: o=airius.com

objectClass: top

objectClass: organization

ACI: (target=“ldap:///o=airius.com”)(targetattr=*)

(version 3.0; acl “acl 1”; allow (all)

groupdn = “ldap:///cn=Directory Administrators, o=airius.com”;)

ACI: (target=”ldap:///o=airius.com”)(targetattr=*)

(version 3.0; acl “acl 2”; deny (all)

groupdn != “ldap:///cn=Directory Administrators, o=airius.com”

and dayofweek = “Sun, Tues, Fri” and

(timeofday >= “0100” and timeofday <= “0300:);)
```

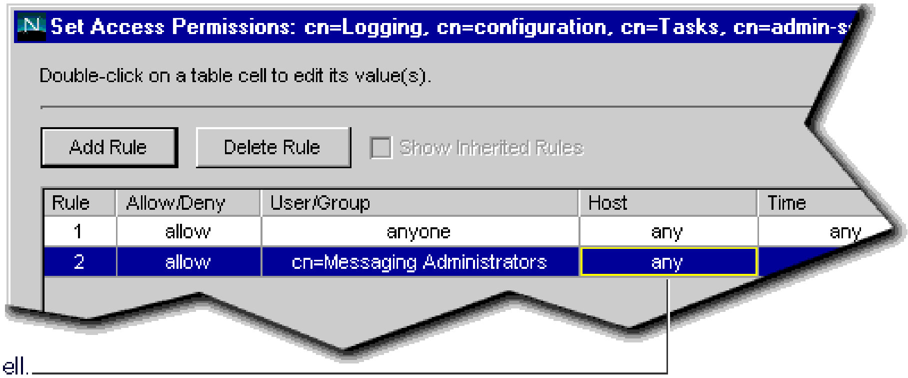
Setting Access Permissions for a Server Task

To set access permission for a server task:

1. In Netscape Console, select a server and open its console.
2. From the server Tasks, select the task you want to allow or deny access permission to.



3. From the Edit menu, choose Set Access Permissions. The ACI Editor appears.



Set Access Permissions: cn=Logging, cn=configuration, cn=Tasks, cn=admin-s
Add Rule
Delete Rule
□ Show Inherited Rules

Double-click on a table cell to edit its value(s).

Rule	Allow/Deny	User/Group	Host	Time
1	allow	anyone	any	any
2	allow	cn=Messaging Administrators	any	

To enter or edit data, double-click a cell.

4. To create a new rule, click Add Rule. A default rule is added to the table.

5. To edit a rule in the table, single-click a cell to edit its contents, or double-click the cell to display a dialog box for entering additional information. Cells and related options are summarized in Table 6.2.

   Continue using the Access Control Editor to create rules and enter settings as necessary, then click Save Changes.

6. Click OK.

7. Restart the server.

Table 6.2 The ACI Editor Settings and Options

Setting	What it does	Options
Rule number	Indicates the order in which the rule was created.	No options available.
User /Group	Designates users or groups to be affected by this rule.	Double-click this cell to display the Select Users and Groups window. Enter information to create a list of users and groups to be affected by this rule. **Add User/Group to List.** Use this pull-down list to indicate whether you're adding a user or group to the list. **Blank Input field.** Enter the full DN for the user or group you want to add to the list. Examples: • uid=JDoe,ou=People,o=mcom.com • cn=Accounting Manager, ou=Group,o=mcom.com **Add.** Adds to the list the user or group you specified in the blank input field (above). **Remove from List.** Removes a selected user or group from the list. **Find Users and Groups.** Displays the Search Users and Groups window so you can locate a user or group you want to add to the list. **All users/groups except those specified in the list.** When checked, excludes the users and groups listed from the rule you create. **Authentication Method.** Choose None if you don't want to use client authentication at all. Choose Simple if you want to use basic user ID/ password authentication. Choose SSL if you want to use SSL certificates for authentication. Choose SASL EXTERNAL if you've written a Directory Server plug-in for use with SASL authentication. **User DN Attribute.** Enter an attribute, such as `manager` or `owner`, that contains a user DN with a value that's subject to change. For example, you can set up an ACI that allows Mary's manager (`manager: uid=asmith`) to access Mary's employment data. When Mary transfers to another department, her DN is changed (`manager: uid=bjones`). The ACI automatically provides rights to her new manager .

Table 6.2 The ACI Editor Settings and Options

Setting	What it does	Options
Host	Designates host computers affected by this rule.	Enter a host name or IP address. You can use wildcards to enter multiple host names at one time. You can only use the wildcard .* and only at the end of an IP address. The * must replace an entire byte in the address. For example, 198.95.251.* is acceptable; 198.95.251.3* is unacceptable.
Time	Specifies an interval when the rule will be in effect.	Enter in 24-hour format (HHMM).
Allow/Deny	Specifies whether to grant or restrict access to the resources named in this rule.	Choose Allow or Deny from the drop-down list.
Rights	Specifies user rights allowed or denied by this rule. (When setting rights for a task, you typically check all of these.)	**Read.** User can view a file. Includes HTTP methods GET, HEAD, POST, and INDEX. **Write.** User can change or delete file. Includes HTTP methods PUT, DELETE, MKDIR, RMDIR, MOVE. **Add.** User can add directory entries. **Delete.** User can delete files. **Search.** Indicates whether data can be searched for. Users must have Search and Read rights in order to view the data returned as part of a search operation. **Compare.** Indicates whether data may be used in comparison operations. With compare rights, the directory returns a yes or no in response to an inquiry, but the user cannot see the value of the entry or attribute. **Selfwrite.** Indicate whether people can add or delete themselves from a group. This right is only used for group management.

Table 6.2 The ACI Editor Settings and Options

Setting	What it does	Options
Check Syntax	Lets you view the ACI syntax as stored in the directory.	You cannot edit the syntax in the Check Syntax dialog box. You must use the Edit Attributes dialog box. In the Access Control Editor, click Edit Attributes.
Edit Attributes	Displays a dialog box for editing ACI search targets.	Use the Edit ACI Attributes dialog box to edit the following: **ACI Name.** Enter a name for the rule you're creating. **Target.** Enter a valid DN to specify the directory entry this rule will apply to. Example: `o=airius.com`. **Target Filter.** Enter a search filter to use to set the rule target. Example: `ou=accounting, ou=engineering`. **Target Attribute.** Specify one or more attributes to which the rule applies. Separate multiple attributes with double vertical bars. Example: `userpassword || telephonenumber.`

Chapter

7

Using SNMP to Monitor Servers

You can use Simple Network Management Protocol (SNMP) together with Netscape management information bases (MIB) and network management software, such as HP OpenView, to monitor your servers in real time just as you monitor other devices in your network. If you're using Windows NT, SNMP service is built in, and you use the Windows NT Control Panel to manage it. If you're using Unix or another platform that doesn't support the SNMP multiplexing protocol (SMUX), you can use Netscape Console to manage the SNMP service Netscape provides.

This chapter contains the following sections:

- SNMP Basics
- Setting Up SNMP on a Netscape Server
- Using a Proxy SNMP Agent
- Reconfiguring the SNMP Native Agent
- Enabling and Starting the SNMP Master Agent
- Configuring the SNMP Master Agent
- Enabling the Subagent

SNMP Basics

SNMP is a protocol used to exchange data about network activity. With SNMP, data travels between a managed device and a network management station (NMS) where users remotely manage the network. A managed device is anything that runs SNMP, such as hosts, routers, and Netscape servers.

An NMS is usually a powerful workstation with one or more network management applications installed. A network management application such as HP OpenView graphically shows information about managed devices. For example, it might show which servers in your enterprise are up or down, or the number and type of error messages received. When you use SNMP with a Netscape server, this information is transferred between the NMS and the sever through the use of two types of agents.

SNMP Subagent

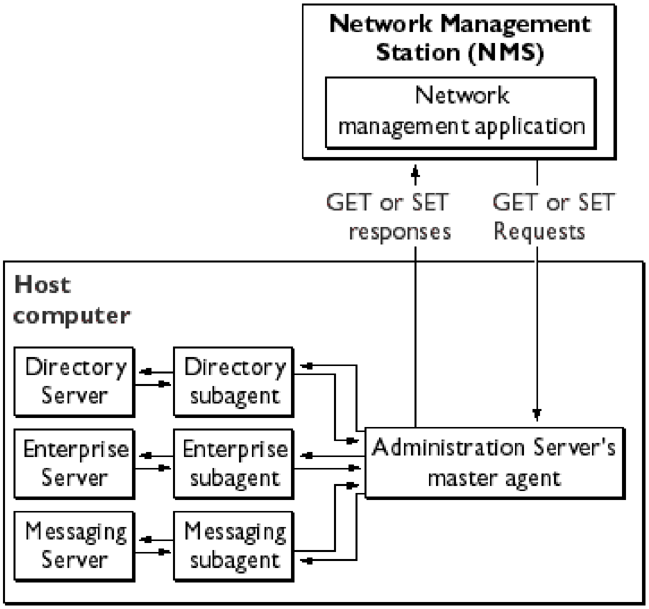
The subagent gathers information about the server and passes the information to the server's master agent. Every Netscape server, except for the Administration Server, has as subagent.

SNMP Master Agent

The master agent exchanges information between the various subagents and the NMS. The master agent is installed with Netscape Console Administration Server.

You can have multiple subagents installed on a host computer, but only one master agent (see Figure 7.1). For example, if you had the Directory Server, the Enterprise Server, and the Collabra Server all installed on the same host, the subagents for each of the servers would communicate with the same master agent.

Figure 7.1 Interaction between the a network management station and a host computer.



How SNMP Works

A managed entity, such as a server, stores variables pertaining to network management. Variables that the master agent can access are known as managed objects. Managed objects are defined in a tree-like hierarchy known as a server's management information base (MIB).

Each Netscape server subagent provides a management information base (MIB) for use in SNMP communication. The MIB is a tree-like hierarchy that contains variables pertaining to the server's management. The server reports significant events to the network management station (NMS) by sending messages or *traps* containing these variables. The NMS can also query the server's MIB for data, or can remotely change variables in the MIB.

Netscape MIBs

Each Netscape server has its own management information base (MIB). All Netscape MIBs are located at

```
<server root>/plugins/snmp
```

A server's MIB contains variable definitions pertaining to network management for that particular server. See your server's *Administrator's Guide* for detailed information about your server's network management variables. Additionally, each Netscape server uses an Administration Server MIB.

The Administration Server MIB

The Netscape Console Administration Server MIB is a file named

`netscape-main.mib`.

This file lists each object identifier for all servers currently supported by Netscape. It also defines the object identifier shared by all Netscape servers as

netscape OJBECT IDENTIFIER: :={enterprises 1450}

Types of SNMP Messages

GET and SET are two types of messages defined by SNMP. GET and SET messages are sent by an NMS to a master agent. You can use one or the other, or both with Netscape Console Administration Server. Messages sent by the server to the NMS are known as traps. The following examples best illustrate the use of GET, SET, and trap messages.

NMS-initiated communication. The NMS either requests information from the server or changes the value of a variable store in the server's MIB. For example:

1. The NMS sends a message to the Administration Server master agent. The message might be a request for data (a GET message), or an instruction to set a variable in the MIB (a SET message).

2. The master agent forwards the message to the appropriate subagent.

3. The subagent retrieves the data or changes the variable in the MIB.
4. The subagent reports data or status to the master agent, then the master agent forwards the message back (a GET message) to the NMS.
5. The NMS displays the data textually or graphically through its network management application.

Server-initiated communication. The server subagent sends a message or trap to the NMS when a significant event has occurred. For example:

1. The subagent informs the master agent that the server has stopped.
2. The master agent sends a message, or trap reporting the event to the NMS.
3. The NMS displays the information textually or graphically through its network management application.

Setting Up SNMP on a Netscape Server

In general, to use SNMP you must have a master agent and at least one subagent installed and running on your system. You need to install the master agent before you can enable a subagent.

The procedures for setting up SNMP are different depending upon your system. Table 7.1 provides an overview of the procedures you follow for various situations. The actual procedures are described in detail later in the chapter.

Before you begin, examine your system.

- Is your system already running an SNMP agent (an agent that's native to your operating system)?
- If so, does your native SNMP agent support SMUX communication? (If you're using the AIX platform, your system supports SMUX.)

See your system documentation for information on how to verify this information.

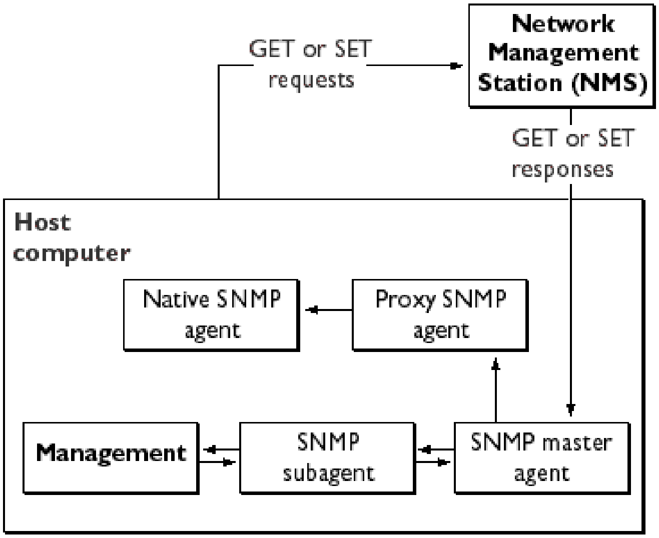
Table 7.1 Overview of procedures for enabling SNMP master agents and subagents.

If your server meets these conditions....	...follow these procedures. These are discussed in detail in the following sections.
No native agent is currently running	**8.** Start the master agent. **9.** Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • No need to continue using native agent	**1.** Stop the native agent when you install the master agent for your Administration Server. **2.** Start the master agent. **3.** Enable the subagent for each server installed on the system.
• Native agent is currently running • No SMUX • Needs to continue using native agent	**1.** Install a proxy SNMP agent. **2.** Start the proxy SNMP agent. **3.** Restart the native agent using a port number other than the master agent port number. **4.** Start the master agent. **5.** Enable the subagent for each server installed on the system.
• Native agent is currently running • SMUX supported	**1.** Reconfigure the SNMP native agent. **2.** Enable the subagent for each server installed on the system.

Using a Proxy SNMP Agent

You need to use a proxy SNMP agent when you already have a native agent running (Figure 7.2), and you want to continue using it concurrently with a Netscape Console master agent. Before you start, be sure to stop the native master agent. (See your system documentation for detailed information.)

Figure 7.2 Using a proxy server when you're running a native SNMP agent.



To use a proxy agent, you'll need to install it and then start it. You'll also have to restart the native SNMP master agent **using a port number other than the one the Netscape Console master agent is running on**.

Installing the Proxy SNMP Agent

To install the SNMP proxy agent, edit the `CONFIG` file (you can give this file a different name), located in `plugins/snmp/sagt` in the server root directory, so that it includes the port that the SNMP daemon will listen to. It also needs to include the MIB trees and traps that the proxy SNMP agent will forward.

Here is an 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;
```

Starting the Proxy SNMP Agent

To start the proxy SNMP agent, at the command prompt, enter:

```
# sagt -c CONFIG&
```

Restarting the Native SNMP Daemon

After starting the proxy SNMP agent, you need to restart the native SNMP daemon at the port you specified in the CONFIG file. To restart the native SNMP daemon, at the command prompt, enter

```
# snmpd -P <port number specified in the CONFIG file>
```

For example, on the Solaris platform, using the port in the previously mentioned example CONFIG file, you'd enter

```
# snmpd -P 1161
```

Reconfiguring the SNMP Native Agent

If your SNMP daemon is running on AIX, it supports SMUX. For this reason, you don't need to install a master agent. However, you do need to change the AIX SNMP daemon configuration.

AIX uses several configuration files to screen its communications. One of them, snmpd.conf, needs to be changed so that the SNMP daemon accepts the incoming messages from the SMUX subagent. For more information, see the online manual page for snmpd.conf. You need to add a line to define each subagent.

For example, you might add this line to the snmpd.conf:

```
smux 1.3.6.1.4.1.1.1450.1 "" <IP_address> <net_mask>
```

IP_address is the IP address of the host the subagent is running on, and net_mask is the network mask of that host.

Note Do not use the loopback address 127.0.0.1; use the real IP address instead.

Enabling and Starting the SNMP Master Agent

Master agent operation is defined in an agent configuration file named CONFIG. You can edit the CONFIG file using Netscape Console, or you can edit the file manually.

Manually Configuring the SNMP Master Agent

To configure the master SNMP agent manually:

1. Log in as root.
2. Check to see if there is an SNMP daemon (snmpd) running on port 161.

   If an SNMP daemon is running, make sure you know how to restart it and which MIB trees it supports. Then kill its process.
3. Edit the CONFIG file located in plugins/snmp/magt in the server root directory.
4. (Optional) Define sysContact and sysLocation variables in the CONFIG file.

Editing the Master Agent Config File

The CONFIG file defines the community and the manager that master agent will work with. The manager value should be a valid system name or an IP address. Here is an example of a basic CONFIG file:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            <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 add initial values for sysContact and sysLocation which specify the sysContact and sysLocation MIB-II variables. Note that the strings for sysContact and sysLocation in this example are enclosed in quotes. Any string that contains spaces, line breaks, tabs, and so on must be in quotes. You can also specify the value in hexadecimal notation.

Here is an example of a CONFIG file with sysContract and sysLocation variables defined:

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            nms2
                   SEND ALL TRAPS TO PORT 162
                   WITH COMMUNITY public

INITIAL            sysLocation “Server room
501 East Middlefield Road
Mountain View, CA 94043
USA”

INITIAL            sysContact “John Doe
email: <jdoe@netscape.com>”
```

Starting the SNMP Master Agent

Once you have installed the SNMP master agent, you can start it manually or by using Netscape Console.

Manually Starting the SNMP Master Agent

To start the master agent manually, enter the following at the command prompt:

```
# magt CONFIG INIT&
```

The `INIT` file is a nonvolatile file that contains information from the MIB-II system group, including system location and contact information. If `INIT` doesn't already exist, starting the master agent for the first time will create it. An invalid manager name in the `CONFIG` file will cause the master agent startup to fail.

To start a master agent on a nonstandard port, use one of two methods:

Method one: In the `CONFIG` file, specify a transport mapping for each interface over which the master agent listens for SNMP requests from managers. Transport mappings allow the master agent to accept connections at the standard port and at a nonstandard port. The master agent can also accept SNMP traffic at a nonstandard port. The maximum number of concurrent SNMP is limited by your target system's limits on the number of open sockets or file descriptors per process. Here is an example of a transport mapping entry:

```
TRANSPORT          extraordinary   SNMP
                   OVER UDP SOCKET
                    AT PORT 11161
```

After editing the `CONFIG` file manually, you should start the master agent manually by typing the following at the command prompt:

```
# magt CONFIG INIT&
```

Method two: Edit the `/etc/services` file to allow the master agent to accept connections at the standard port as well as at a nonstandard port.

Starting the SNMP Master Agent Using Netscape Console

To start the SNMP master agent using Netscape Console:

1. Log in as root.
2. In Netscape Console, open the console for the Administration Server that is running the management software.
3. In the Administration Server Console, choose Tasks, then double-click Configure SNMP Master Agent.
4. Click Start.

Configuring the SNMP Master Agent

Once you've enabled the master agent and enabled a subagent on a host computer, you need to configure the host's Administration Server. This entails specifying community strings and trap destinations.

Configuring the Community String

A community string is a text string that an SNMP agent uses for authorization. This means that a network management station would send a community string with each message it sends to the agent. The agent can then verify whether the network management station is authorized to get information. Community strings are not concealed when sent in SNMP packets; strings are sent in ASCII text.

The master agent uses the community string for authentication.You can configure the community string for the SNMP master agent from Netscape Console. You also define which SNMP-related operations a particular community can perform. From Netscape Console, you can also view, edit, and remove the communities you have already configured.

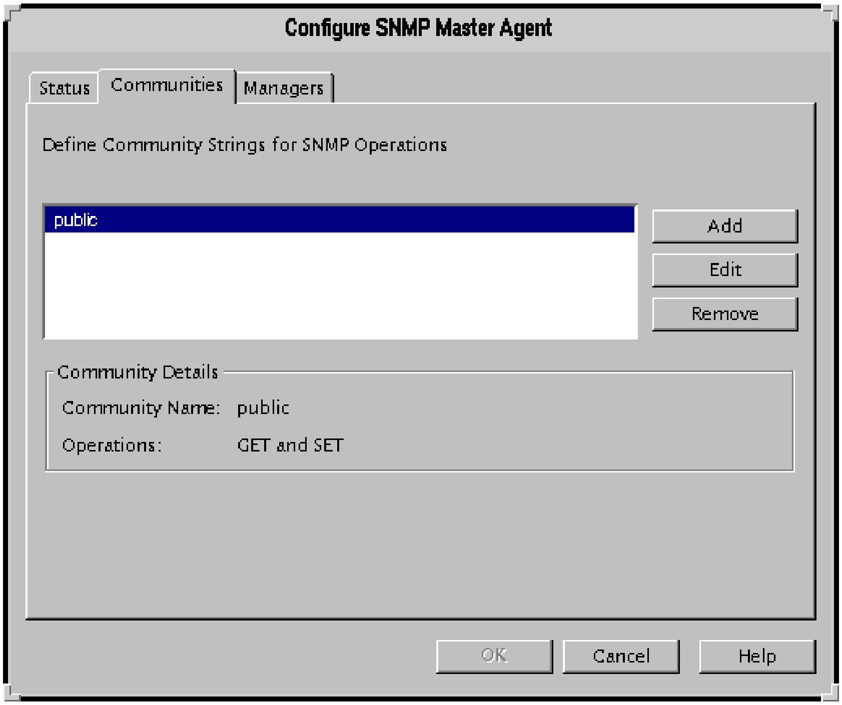
Adding, Editing, or Removing a Community String

To add, modify, or remove a community string:

1. In Netscape Console, open the console for the Administration Server that is running the management software.

2. In the Administration Server Console, click Tasks.

3. Click the Configure SNMP Master Agent button, then click Communities.



Define Community Strings for SNMP Operations

public

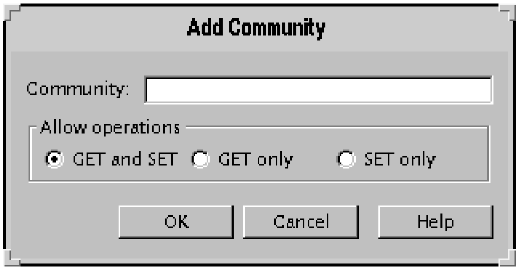
Community Details

Community Name: public

Operations: GET and SET

4. Click Add, Edit, or Remove as necessary.

5. Enter community string information as necessary:



Add Community

Community:

Allow operations

● GET and SET ○ GET only ○ SET only

Community. Enter a community string you want to add or edit. A community string is a password that an SNMP agent uses for authorization.

GET and SET. Choose this option if you want to use this community string for requesting data or replying to messages, and for setting variable values.

GET only. Choose this option if you want to use this community string only for requesting messages or replying to messages.

SET only. Choose this option if you want to allow this community string only for setting variable values.

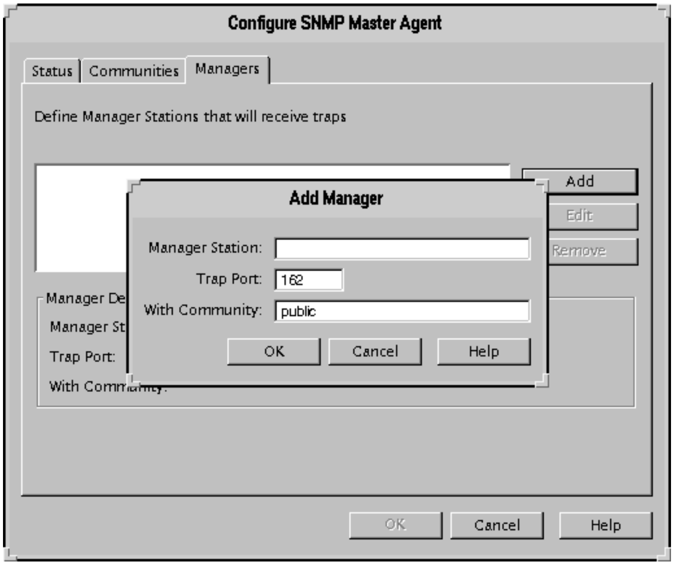
6. Click OK.

Configuring Trap Destinations

An SNMP trap is a message the SNMP agent sends to a network management station. For example, an SNMP agent sends a trap when an interface's status has changed from up to down. The SNMP agent must know the address of the network management station so that it knows where to send traps. You can configure this trap destination for the SNMP master agent from Netscape Console. You can also view, edit, and remove the trap destinations you have already configured. When you configure trap destinations using Netscape Console, you are actually editing the `CONFIG` file.

To Add, Edit, or Remove a Trap Destination:

1. In Netscape Console, open the console for the Administration Server that is running the management software.
2. In the Administration Server Console, Click Tasks.
3. Click the Configure SNMP Master Agent button, then click Managers.



Manager Station:

Trap Port: 162

With Community: public

4. Click Add, Edit, or Remove as necessary.
5. Enter Manager information as necessary:

   Manager Station. Enter a valid system name or an IP address for the NMS.

   Trap Port. Enter the port number the NMS uses to listen for traps. (The default is 162.)

   With Community. Enter the community string you want to use in the trap. A community string is a password that an SNMP agent uses for authentication

6. Click OK.

Enabling the Subagent

For information on enabling the subagent, see the *Administrator's Guide* for your Netscape server. If you need more information, see your system documentation.

3

Administrator's Guide to Netscape Administration Server

Chapter

8

Administration Server Basics

The Administration Server processes requests from the servers in a server group, then invokes the programs required to fulfill the servers' requests. See "The Administration Server" on page 16 for a brief overview of Netscape Console architecture.

This chapter contains the following sections:

- Starting the Administration Server
- Stopping the Administration Server
- Logging Options
- The Administration Page

Starting the Administration Server

The Administration Server automatically starts once it's installed. When you need to restart the server, you can start it from Netscape Console or from the command line.

From the Command Line

To restart the Administration Server from the command line:

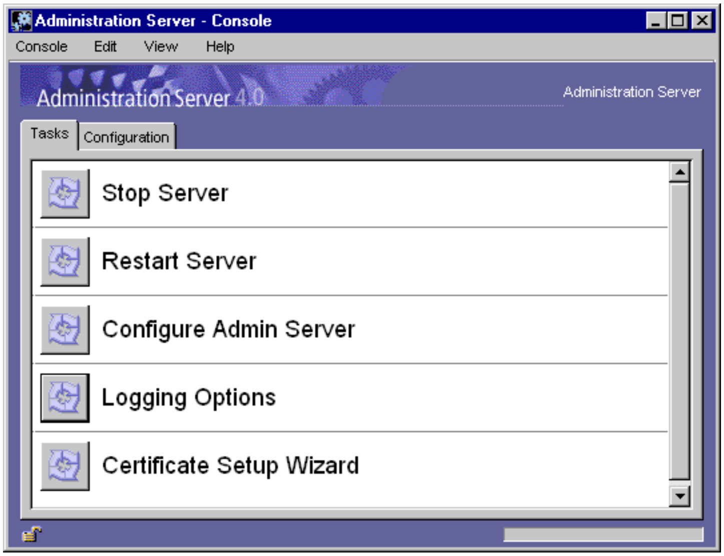
Unix. In the server root, enter `./start-admin`.

Windows NT. From the Start Menu, choose Run. Then enter `<server root>/start-admin.cmd` .

From Netscape Console

To restart the Administration Server using Netscape Console:

1. From the Navigation Tree in Netscape Console, open the Administration Server you want to start.
2. Click Tasks, then choose Restart Server.



Stopping the Administration Server

To stop the Administration Server:

1. From the Navigation Tree in Netscape Console, open the Administration Server you want to stop.
2. Click Tasks, then choose Stop Server.

Logging Options

Log files can help you monitor the Administration Server's activity, and can also help you troubleshoot server problems. Server logs use the Common Logfile Format, a commonly supported format that provides a fixed amount of information about the server.

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 in to the server. By default, the file is located at `admin-serv/logs/error`.

To set a new path for Administration Server log files:

1. In the Navigation Tree of Netscape Console, select an Administration Server and open its console.
2. Double-click Logging Options
3. In the Logging Options window, enter new paths as necessary:

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

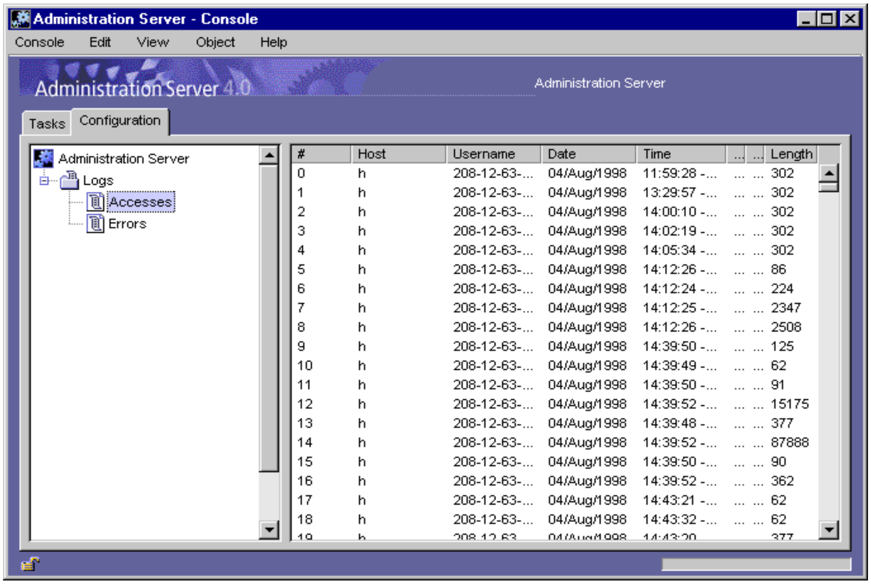
   Error Log - Log File. Enter the path to the directory where you want the Administration Server to store the error log file.

4. Click OK.

Viewing the Access Log

To view the access log:

1. In the Navigation Tree of Netscape Console, select an Administration Server and open its console.
2. Click Configuration.
3. In the configuration tree, click + to expand the Logs directory, then click the Accesses icon.



#	Host	Username	Date	Time	...	...	Length
0	h	208-12-63-...	04/Aug/1998	11:59:28 -...	...	...	302
1	h	208-12-63-...	04/Aug/1998	13:29:57 -...	...	...	302
2	h	208-12-63-...	04/Aug/1998	14:00:10 -...	...	...	302
3	h	208-12-63-...	04/Aug/1998	14:02:19 -...	...	...	302
4	h	208-12-63-...	04/Aug/1998	14:05:34 -...	...	...	302
5	h	208-12-63-...	04/Aug/1998	14:12:26 -...	...	...	86
6	h	208-12-63-...	04/Aug/1998	14:12:24 -...	...	...	224
7	h	208-12-63-...	04/Aug/1998	14:12:25 -...	...	...	2347
8	h	208-12-63-...	04/Aug/1998	14:12:26 -...	...	...	2508
9	h	208-12-63-...	04/Aug/1998	14:39:50 -...	...	...	125
10	h	208-12-63-...	04/Aug/1998	14:39:49 -...	...	...	62
11	h	208-12-63-...	04/Aug/1998	14:39:50 -...	...	...	91
12	h	208-12-63-...	04/Aug/1998	14:39:52 -...	...	...	15175
13	h	208-12-63-...	04/Aug/1998	14:39:48 -...	...	...	377
14	h	208-12-63-...	04/Aug/1998	14:39:52 -...	...	...	87888
15	h	208-12-63-...	04/Aug/1998	14:39:50 -...	...	...	90
16	h	208-12-63-...	04/Aug/1998	14:39:52 -...	...	...	362
17	h	208-12-63-...	04/Aug/1998	14:43:21 -...	...	...	62
18	h	208-12-63-...	04/Aug/1998	14:43:32 -...	...	...	62
19	h	208-12-63	04/Aug/1998	14:43:20			377

Viewing the Error Log

To view the error log:

1. In the Navigation Tree of Netscape Console, select an Administration Server and open its console.
2. Click Configuration.
3. In the configuration tree, click + to expand the Logs directory, then click the Errors icon.

The Administration Page

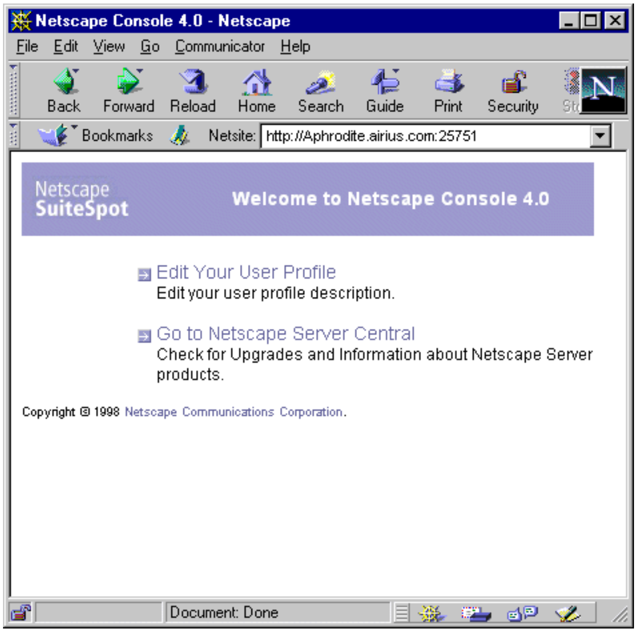
The Administration page provides links to sites or services of interest to system administrators. For example, in Figure 8.1 the Administration page contains links that allow an administrator to: change an end user's profile or access a web site for downloading server software.

To access the Administration page:

Open a browser, and enter the qualified host name and port number for the Administration Server you want to access.

Example: `Aphrodite.airius.com:26751`

Figure 8.1 The Administration page is typically customized to meet a company's specific needs.


Netscape Console 4.0 - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security
Bookmarks
Netsite:

http://Aphrodite.airius.com:25751

Netscape SuiteSpot

Welcome to Netscape Console 4.0

Edit Your User Profile
Edit your user profile description.

Go to Netscape Server Central
Check for Upgrades and Information about Netscape Server products.

Copyright © 1998 Netscape Communications Corporation.

Document: Done

Chapter 9

Administration Server Configuration

This chapter describes the configuration options you can use with the Administration server.

This chapter contains the following sections:

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

Network Settings

Network settings affect the way the Administration Server runs. You can change the system user account that runs the Administration Server. This is a user account you set up with your computer's operating system. (By default, the user is `nobody` on Unix, and `LocalSystem` on Windows NT.)

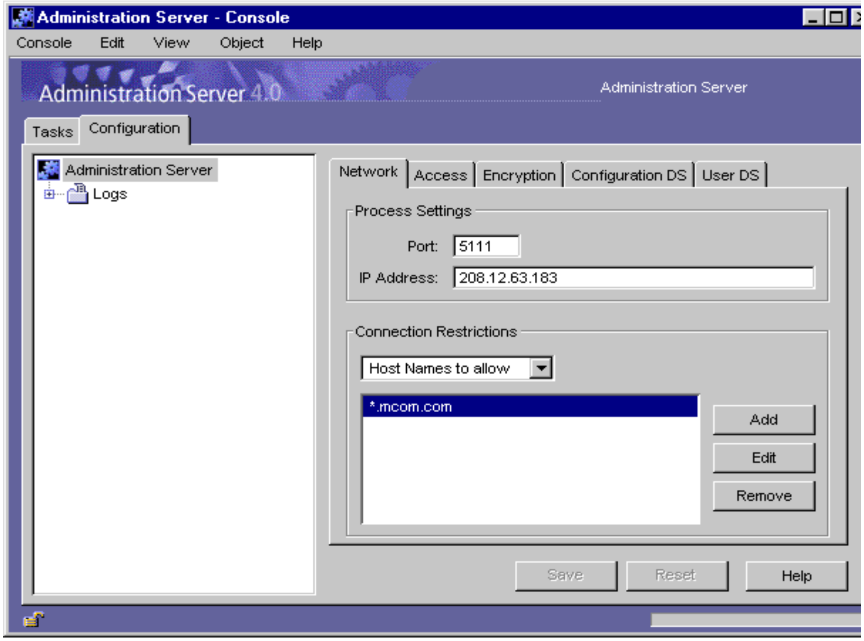
You can change the port number that the Administration Server listens to. The port number can be any number between 1 and 65535, but it is typically a random number greater than 1024. For security reasons, consider changing the port number regularly.

You can change the IP address for a server. This is useful if the host system is connected to multiple networks and you want to specify a single IP address the server should use for incomming requests and connections.

You can also specify which hosts are allowed to connect to the Administration Server.

To configure Administration Server network settings:

1. In Netscape Console, select the Administration Server you want to modify, and then click Open.
2. Click the Configuration tab, and then click Network.



Administration Server - Console

Process Settings

Port: 5111

IP Address: 208.12.63.183

Connection Restrictions

Host Names to allow

*.mcom.com

3. Enter network settings:

   Port. Enter the port number you want the Administration Server to use. The port number can be any number between 1 and 65535, but it is typically a random number greater than 1024.

   IP Address. Enter the IP address you want the server to use for incoming requests and connections.

Server UID. Enter the system user account you want to use to run the Administration Server.

Connection Restrictions. Displays a list of hosts currently allowed to connect to the Administration Server. Use the drop-down list to indicate 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. Using IP addresses may provide faster authentication.

Add. Displays a dialog box for adding a host to the list of computers allowed to connect to the Administration Server.

Edit. Displays a dialog box for editing a Host IP address or DNS name on the list of computers allowed to connect to the Administration Server.

Remove. Removes a selected entry from the list of allowed hosts.

4. Click OK.

Access Settings

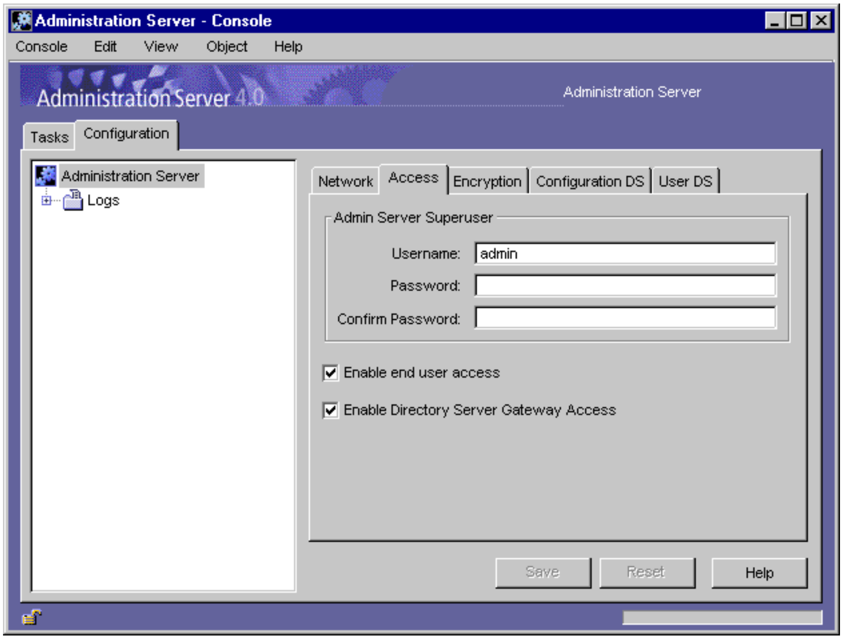
Access settings specify who is allowed to access these areas of the Administration Server:

- Server group management - This includes all server management tasks such as starting, stopping, and configuring the servers in a group.
- End-user page - This is the Administration Server html page that end users typically access to either modify their own user data, or to download shared software. See "End-User Access to the User Directory" on page 46 for more information.
- Directory Server Gateway - The Directory Server Gateway is a service that provides web-based access to the entire user dirctory. The Directory Server Gateway must be installed before you can use this option. See the *Administrator's Guide to Directory Server 4.0* for more information.

To set access settings for the Administration Server:

1. In Netscape Console, select the Administration Server you want to modify, and then click Open.

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



Admin Server Superuser

Username: admin

Password:

Confirm Password:

☑ Enable end user access

☑ Enable Directory Server Gateway Access

3. Enter access information:

   User name. Enter Netscape Console Administrator user ID. This is the user listed in the file <`server_root>/admin-serv/config/admpw.` This is the user name you entered during installation. This user has full access to all features in the Administration Server.

   Password. Enter Netscape Console Administrator's password.

   Confirm Password. Enter Netscape Console Administrator's password again to confirm it.

   Enable end-user access. Select this option if you want to allow end users to access the end-user page. Users will be able to access the end-user page using the same URL that administrators do. But they will only see a single form with their user information. An end user can change his or her own password or update any other information stored in his or her own entry in the user database.

Enable Directory Server Gateway Access. By default, this option is selected for you. Deselect it to disable access to the Directory Server Gateway.

4. Click OK.

Encryption Settings

All Netscape 4.0 servers support the SSL protocol and PKCS #11 APIs for encryption communication. Encryption prevents communication between the Administration Server and other servers from eavesdropping and tampering. You need to configure the Administration Server for SSL if it will communicate with SSL-enabled servers.

Before you can use SSL with the Administration Server, you must first enable and activate SSL on the server. The Certificate Setup Wizard in Netscape Console simplifies the enabling process for you. The following procedures walk you through using the Certificate Setup Wizard, and then activating SSL on the Administration Server.

Enabling SSL on a 4.x Administration Server

To enable SSL on a 4.x Administration Server:

1. In Netscape Console, in the navigation tree, select the Administration Server instance you want to use SSL encryption with.
2. Click Open Server to open the Administration Server Console.
3. In the Administration Server Console, from the Console menu, choose Certificate Setup Wizard.

4. Provide information as prompted. See “Obtaining and Installing a Certificate” on page 67 for detailed information.

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

Activating SSL on a 4.x Administration Server

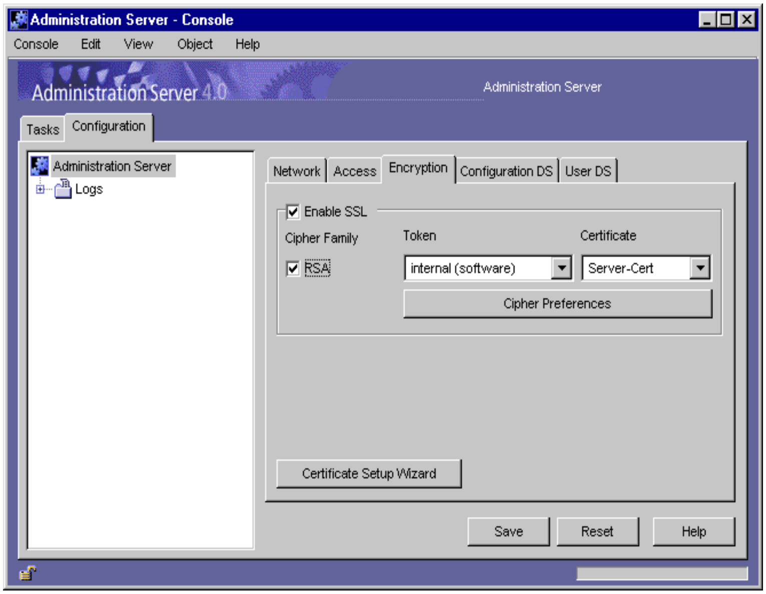
The cipher family and preferences you specify here are used to provide SSL communication between Administration Server and Netscape Console.

To activate SSL on a 4.x Administration Server:

1. In Netscape Console, in the navigation tree, select the Administration Server instance you want to use SSL encryption with.

2. Click Open Server to open the Administration Server console, and then click Configure the Administration Server.

3. In the Configuration window, click Encryption.



☑ Enable SSL

Cipher Family: ☑ RSA

Token: internal (software)

Certificate: Server-Cert

4. Enter information as appropriate:

   Enable SSL. Choose this option if you want to secure your enterprise with Secure Sockets Layers (SSL) encryption. The following are enabled only when you turn on SSL Encryption:

   Cipher Family. When you enable SSL Encryption, the cipher families available to you are listed here. Select the cipher families you want to use.

   Token to Use. Choose Internal (Software-based) if the key is stored in the local key database. All other choices available to you on this list are device-based. This means the key is stored on an external device such as a Smart Card.

   Certificate. Certificate information is stored in the certificate database. If you're unsure of the Certificate to use, view the Certificate Management dialog for more information. To view the Certificate Management dialog, from the File menu, choose Certificate Management.

   Cipher Preferences. A cipher is the algorithm used in encryption. This list displays the cipher preferences you've selected.

5. Click OK.

Directory Settings

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

The Configuration Directory

When you install a server, you're asked for the location of the Directory Server that will store your server's configuration data. The Directory Server you specify contains the default *configuration directory*. The configuration directory is a subtree of the Directory Server. Data such as network topology information, console configuration, and server instance entries (SIEs) are stored in this subtree. Each time you install a server or change its configuration, the changes are stored in this subtree. For example, when you change a server's port number or turn on SNMP, the relevant data is stored in the configuration directory of the Directory Server.

Changing the Configuration Directory Server

You can designate a different host or port number for the configuration Directory Server.

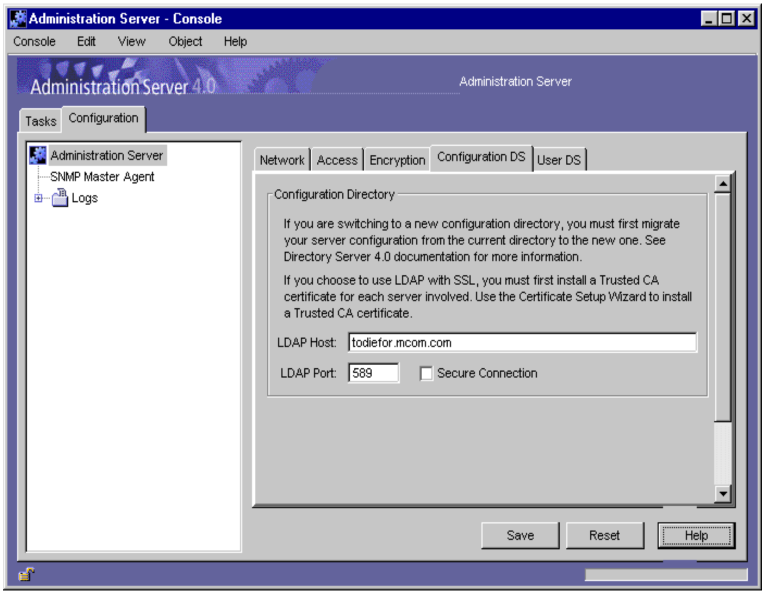
Note Changing the configuration Directory Server has serious and far-reaching impacts on the rest of the servers in the server group! If you change a setting here, you must make the same change in every server in the administration domain.

To change the configuration Directory Server settings:

1. In Netscape Console, choose an Administration Server and open it.

2. Click Configuration.

3. Click Configuration DS.



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 4.0 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: todiefor.mcom.com

LDAP Port: 589 □ Secure Connection

4. Modify settings as appropriate.

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

   LDAP Port. Enter the port number for the configuration directory this Administration Server uses.

   Use SSL. Select this option if the new configuration directory is already SSL enabled.

5. Click Save.

The User Directory

The user directory is a subtree of the Directory Server. It uses a suffix that you create, such as `o=airius.com`. The user directory is used for authentication and for local server management. It stores all user and group data, accounts 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 with each of five Mail Servers.

User Directory Settings

When you're installing a Netscape server, you are prompted to specify a user directory that is associated with the administrative domain. By default, a server group uses the same user directory associated with its domain. Also by default, an individual server uses the same user directory as its server group. There may be times when you need to override default user directory settings at the server, 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. Or you might want to temporarily change the user directory for a server group when you're testing a new Directory Server for the group, and you don't want to impact your existing user directory.

User Authentication and Directory Failover Support

When a user logs in to Netscape Console, he enters his user ID which is checked against the user directory. If the user ID cannot be authenticated in a user directory, the user cannot successfully log in to Netscape Console.

If you're using a Netscape Console 4.1 version or higher, you can list more than one user directory that can be used for authenticating users IDs. This is useful when the Directory Server that contains your primary user directory is not running or is not accessible. If the user directory has been replicated in other host locations, Netscape Console continues to check the user ID against each user directory in the list until authentication can be made.

To list user directories to be used for failover support, follow instructions for "Changing User Directory Settings for a Domain" on page 143 or "Changing User Directory Settings for a Server Group" on page 144. For information on replicating the user directory, see the *Directory Server 4.0 Administrator's Guide.*

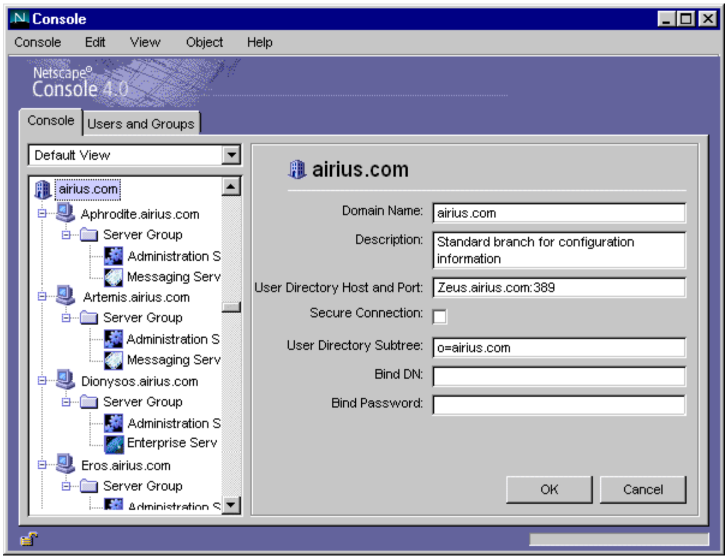
Changing User Directory Settings for a Domain

You must be the Configuration Administrator or Domain Administrator to change the user directory settings for a domain.

Changing these settings will have serious and far-reaching impacts on the rest of the servers in the domain! If you make changes here, you must restart all the servers in the domain.

To change the user directory settings for a domain:

1. In Netscape Console, select a domain, then click Edit.



airius.com

Domain Name: airius.com

Description: Standard branch for configuration information

User Directory Host and Port: Zeus.airius.com:389

Secure Connection: □

User Directory Subtree: o=airius.com

Bind DN:

Bind Password:

2. Modify domain information as appropriate.

   Domain Name. Enter a fully qualified domain name.
   Example: `airius.mcom.com`

   Description. Enter a name 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. For authentication purposes, you can enter more than one user directory location separated by spaces.

Example:

```
Eros.Airius.com:389 Zeus.Airius.com:389
```

See “User Authentication and Directory Failover Support” on page 142 for more information.

If you specify more than one host computer, each one must be configured identically regarding the following settings:

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

User Directory Subtree. Enter the location of the new user directory. Example: o=mcom.com

Bind DN. Enter the distinguished name for a user who has access permisions to the new user directory. Example: `uid=ginac, ou=people, o=Airius.com.`

Bind Password. Enter the password of the user above.

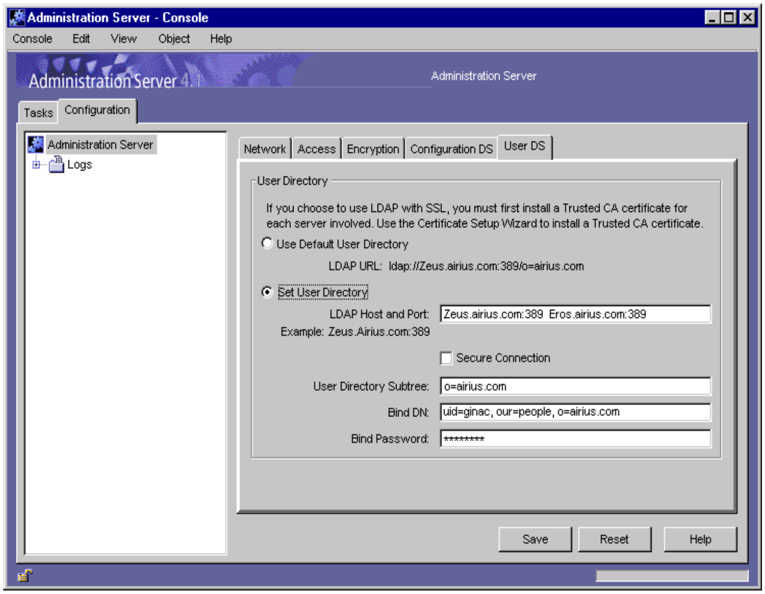
3. Click OK.

Changing User Directory Settings for a Server Group

To change the user Directory Server settings for a server group:

1. In Netscape Console, choose an Administration Server and open it.
2. Click Configuration.

3. Click User DS.



User Directory

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.

○ Use Default User Directory

LDAP URL: ldap://Zeus.airius.com:389/o=airius.com

● Set User Directory

LDAP Host and Port: Zeus.airius.com:389 Eros.airius.com:389

Example: Zeus.Airius.com:389

□ Secure Connection

User Directory Subtree: o=airius.com

Bind DN: uid=ginac, our=people, o=airius.com

Bind Password: ********

4. Modify settings as appropriate.

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

Set User Directory. Choose 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:

```
Eros.Airius.com:389 Zeus.Airius.com:389
```

See "User Authentication and Directory Failover Support" on page 142 for more information

If you specify more than one host computer, each one must be configured identically regarding the following settings:

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

User Directory Subtree. Enter the location of the new user directory. Example: o=mcom.com

Bind DN. Enter the distinguished name for a user who has access permisions to the new user directory. Example: `uid=ginac, ou=people, o=Airius.com.`

Bind Password. Enter the password of the user above.

5. Click OK.

Changing User Directory Settings for A Server

See the server's *Administrator's Guide* for detailed information.

4

Appendixes

Appendix

A

Distinguished Name Attributes and Syntax

Attributes

Distinguished Name (DN) attributes uniquely identify a user or group so that it can be located in the directory server. A DN customarily contains at least three attributes:

- a user’s name or user ID
- an organization name
- a country designation

Most companies use many more attributes in order to store additional user and group information. For example, the DNs for three employees or users in the same company might look like this:

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

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

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

In these examples, all three users work in different departments (ou) for the same company (o), Klondike Corp. The third user works in a different state (st) than the first two users.

These and other common attributes are summarized in this table:

Table A.1 Frequently Used Attributes for Distinguished Names

Attribute Name	Syntax	Description

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

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

email address	mail	User's or group's email address.
given name	givenName	User's first name.

locality	l	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

organization	o	Organization to which the user or group belongs.
		Examples:
		o=Netscape Communications Corp.
		o=Public Power & Gas

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

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

password	userPassword	Password created by a user.

Table A.1 Frequently Used Attributes for Distinguished Names

Attribute Name	Syntax	Description
street	streetAddress	Street number and address of user or group defined by the entry. Example: street=494 Rice Creek Terrace
surname	sn	User's last name.
telephone	telephoneNumber	User's or group's telephone number.

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

user ID	uid	Name that uniquely identifies the person or object defined by the entry.

You can use or create whatever attributes you want to use to meet your company's needs. However, the attributes you use ultimately depend upon how your directory is set up. **All attributes you specify when using the Netscape Console must be identical to the attributes used by your directory server.** See your directory server documentation for information on setting up your directory.

DN Guidelines and Syntax

As you create and modify DNs, you should follow these guidelines:

Separate attributes with a comma. If a distinguished name contains a comma, then the part of the name that uses the comma must also be enclosed in double-quotation marks. For example, to include the string Ace Industry, Corp in a distinguished name, use the form

```
o=”Ace Industry, Corp”, c=US
```

Attributes must match directory schema. If you are using the Netscape Directory Server and schema checking is turned on, then use attributes that can be recognized by the Directory Server and are allowed by the entry's object classes.

Specify attributes in the same sequence or path. Remember that a DN represents a path through a directory tree. For example, the directory server does not recognize these two entries as representing the same user:

```
cn=Ralph Swenson, ou=Accounting, o=Ace Industry, c=US
```

```
cn=Ralph Swenson, o=Ace Industry, ou=Accounting, c=US
```

The organizational unit (ou) and organization (o) attributes are listed in a different sequence.

User ID must be unique. If duplicate user IDs exist in your directory, the affected users will not be able to authenticate to the directory. If you use the `ldapmodify` command line utility to create a user, the utility will not check for duplicate user IDs.

Appendix

B

Administration Server Command Line Tools

The Administration Server bundles the following command line tools:

- `admconfig`
- `admin_ip.pl`
- `ldapsearch` and `ldapmodify`
- `sec-migrate`
- `modutil`

admconfig

The `admconfig` command allows you to configure the Administration Server using the command line instead of using the Netscape Console graphical interface. Use `admconfig` to modify network, access, encryption, or directory settings. It is stored at `<server_root>/bin/admin`.

Syntax

```
admconfig [options] <task> [args] [<task> [args]...]
```

Examples

- This example changes the port number to 33333 and restarts the Administration Server. The verbose level option is set to 5.

  ```
  % admconfig -server jaffer.mcom.com:22222 -user
  phlee:password -verbose 5 -setPort 3333 -restart
  ```

- This example retrieves the hosts from which connections are allowed. The verbose level option is set to 9.

  ```
  % admconfig -ser jaffer.mcom.com:33333 -u
  phlee:password -verb -geth
  ```

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

  ```
  % adminconfig -h r
  ```

Options

You can specify options using the terse single character form, such as `-u` if applicable, or using the longer but more descriptive form such as `-user`. The complete option name does not have to be specified. For example, `-us` will work just as well as `-user`. Just make sure you provide enough letters to ditinguish that option name fromall other options or tasks. Options are not case sensitive. For example, `-USER` and`-User` are both accepted as the `-user option`.

Table B.1 Options You Can Use with admconfig

Option	Description
-con[tinueOnError]	Finish the remaining tasks even when an error occurs. Default behavior is to quit when any task fails without running the remaining tasks.
-enc[ryption]	Use encrypted protocol (https) to connect to the server. The default protocol is http.
-h[elp] [<task>]	Display the usage information [for task].

Table B.1 Options You Can Use with admconfig

Option	Description
-i[nputFile] <filename>	Get options from the specified file. The same options can also be specified on the command line along with the option file. In this case, the options on the command line override the options in the file. The `-inputFile` option within the option file is ignored to prevent recursion.
-ser[ver] [<host>]:<port>	Connect to the server on host at specified port. If no host is specified, the lcal host is used. The server port number (preceded by the colon) is required.
-u[ser] [<uid>]:[<pwd>]	Connect to the server using username and password. If the user name is not specified, the user's login is prompted for the password. The password is echoed back on the user's screen, so if security is a concern, the `-inputFile` option should be used to provide the username and password in a file with suitable permissions. Note that if the `-user` option is specified, then at minimum, the colon must be specified. If the `-user` option is not specified, then the user is prompted for both the username and password.
-verb[ose] [<0-9>]	Set the level of screen output (9=full output, 0=no output).The default level is 5.
-ver[sion]	Display the version and copyright information.

Tasks

You can specify a task using the abbreviated name such as `-r` for restarting the server, or use the complete name `-restart`. Specify a task name that is unique among other tasks or options. The task key words are not case sensitive. Examples: both `-RESTART` and `-Restart` are accepted as the -restart task.

Multiple tasks can be run from the same invocation of `admconfig`. Tasks specified in an input file are run first. Tasks are run in the order specified in the input file and command line.

Table B.2 Tasks You Can Perform with admconfig

Task	Description
`-countA[ccessLogEntries]`	Count the number of entries in the access log file. This task should be performed prior to -view A[ccesslogEntries] in order to determine the number of entries that can be viewed in the access log.
`-viewA[cessLogEntries]`	View the specified entries in the error log file. Syntax: `admconfig [options] -viewA[cessLogEntries] \”<start> <stop>\”` Required parameter includes: `<start>` The first log entry number to start displaying from. `<stop>` The last log entry number to display. The backslash character is required before the quotes surrounding the two arguments to `-viewA[cessLogEntries]`. If the backslash is not provided, on UNIX systems the shell will evaluate the quotes and pass the arguments without the quotes to the command line. This will result in only `<start>` being assigned as the parameter or `-viewA[cessLogEntries]`. The backslash character before the quotes will prevent the quotes from being evaluated and allow both arguments to be assigned as the parameter to `-viewA[cessLogEntries]`s.
`-countE[rrorLogEntries]`	Count the number of entries in the error log file. This task should be performed prior to `-viewErrorLogEntries` in order to determine the number of entries that can be viewed in the error log.

Table B.2 Tasks You Can Perform with admconfig

Task	Description
-viewE[rrorLogEntries]	View the specified entries in the error log file. Syntax: admconfig [options] -viewE[rrorLogEntries] \”<start> <stop>\” Required parameter includes: <start> The first log entry number to start displaying from. <stop> The last log entry number to display. The backslash character is required before the quotes surrounding the two arguments to -viewErrorLogEntries. If not provided, on UNIX systems, the shell will evaluate the quotes and pass the arguments without the quotes to the command line. This will result in only <start> being assigned as the parameter or -viewErrorLogEntries. The backslash character before the quotes will prevent the quotes from being evaluated and allow both arguments to be assigned as the parameter to -viewErrorLogEntries.
-enableD[SGWAccess]	Enable Directory Server Gateway access to the Administration Server.
-disableD[SGWAccess]	Disable Directory Server Gateway access to the Administration Server.
-enablE[ndUserAccess]	Enable end user access to the Administration Server.
-disableE[ndUserAccess]	Disable end user access to the Administration Server.
-getAc[cessLog]	Get the name of the server access log file
-setAc[cessLog]	Set the name of the server access log file. Required parameter includes: <filename> New server access log file.
-getAdd[resses]	Get the addresses from which connections are allowed.
-setAdd[resses]	Set the addresses from which connections are allowed. Required parameter includes: <addresses> New addresses from which connections are allowed.
-getAdminUI[D]	Get the administrator’s user name.

Table B.2 Tasks You Can Perform with admconfig

Task	Description
`-setAdminUI[D]`	Set the administrator's user name Required parameter includes: `<uid>` The new user ID for the administrator.
`-setAdminP[wd]`	Set the administrator's password to the specified value. Required parameter includes: `<password>` The new user password for the administrator.
`-getAdminUs[ers]`	Get the name of the `adminusers` file.
`-setAdminUs[ers]`	Set the name of the `adminusers` file. Required parameter: `<adminusers>` New name for the `adminusers` file.
`GetCa[cheLifetime]`	Get the amount of time that the user authentication is cached.
`-setCa[cheLiftetime]`	Set the amount of time to cache the user authentication. Required parameter includes: `<mesc>` New cache lifetime in mescs.
`-getCl[assname]`	Get the Java `classname` for the Administration Server.
`-setCl[assname]`	Set the Java `classname` for the Administration Server.
`-getDe[faultAcceptLanguage]`	Get the `defaultacceptlanguage`.
`-setDe[faultAcceptLanguage]`	Set the `defaultacceptlanguage`. Required parameter: `<language>` New default accept language.
`-getDS[Config]`	Retrieve the current LDAP server host, port, base DN, and whether the LDAP server is running SSL.

Table B.2 Tasks You Can Perform with admconfig

Task	Description
`-setDS[Config]`	Set the LDAP server host, port, base DN, and whether the LDAP server is running SSL. Syntax: `admconfi [options] -setDS[Config] \"<host> <port> <baseDN> <ssl>\"` Required parameter includes: `<host>` The LDdAP Server host name. `<port>` The LDAP Server port number. `<baseDN>` The LDAP Server base DN. `<ssl> “true” | “false”` depending on whether to use the Secure Sockets Layer to communicate with the LDAP Server. The backslash character is required before the quotes surrounding the four arguments to `-setDS[Config]`. If the backslash is not provided, on Unix systems the shell will evaluate the quotes and pass the arguments without the quotes to the command line. This will result in only `<port>` being assigned as the parameter or `-setDS[Config]`. The backslash character before the quotes will prevent the quotes from being evaluated and allow both arguments to be assigned as the parameter to`-setDS[Config]`.
`-getU[GDSConfig]`	Retrieves the current user/group LDAP server information, including the host, port, base DN, and authentication DN.

Table B.2 Tasks You Can Perform with admconfig

Task	Description

-setU[GDSConfig] [\"<host> <port>
<baseDN> <ssl>
<uid> <pwd>\"]
 	
Optional arguments include:
 	
<host> . The user/group LDAP Server host name.
 	
<port> . The user/group LDAP Server port number.
 	
<baseDN> . The user/group LDAP Server base DN.
 	
<ssl> "true" | "false" . Indicates whether to use the Secure Sockets Layer to communicate with the LDAP Server.
 	
<uid>. Authentication DN used to bind to LDAP Server.
 	
<pwd> . Authentication password used to bind to LDAP Server.
	Sets the user/group LDAP server host, port, baseDN, authentication DN, and authentication password.
	
	You can invoke -setUGDSConfig either with or without parameters. If this task is invoked without any arguments, for example:
	
	`% admconfig -server jaffer.mcom.com:22222 -user`
	`admin:password`
	`-setUGDSConfig`
	
	Then the directory server configuration is reset to the installation defaults. On the other hand, if the task is invoked with all six arguments (all six arguments are required), then they override the installation defaults or the previous values that may have been set.
	
	The backslash character is required before the quotes surrounding the six arguments to `-setUGDSConfig`. If not used, on Unix systems, the shell will evaluate the quotes and pass the arguments without the quotes toadmconfig. This will result in only `<host>` being assigned as the parameter to `-setUGDSConfig`, which will cause the task to fail due to missing arguments.
	
	The backslash character before the quotes will prevent the quotes from being evaluated and allow all arguments to be assigned as the parameter to `-setUGDSConfig`.
	
	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.
	
	IMPORTANT NOTE: The space character is used to parse these six arguments. Therefore, none of the arguments may have spaces in them. To support spaces for arguments such as `<baseDN>`, `<uid>`, and `<pwd>`, the parsing function for these three arguments uses a character replacement scheme to allow spaces to be specified. Simply, whenever a space is required, the + character should be used to indicate to the parser to convert the character to space. For example, to specify `cn=directory manager` as the `<uid>`, users must type `cn=directory+manager`. Because the + character is used in place of the space character, the + character cannot be used as an actual value.

Table B.2 Tasks You Can Perform with admconfig

Task	Description
`-getE[rrorLog]`	Get the name of the server error log file.
`-setE[rrorLog]`	Set the name of the server error log file. Required parameter: `<filename>` New server error log file.
`-getH[osts]`	Get the hosts from which connections are allowed.
`-set[Hosts]`	Set the hosts from which connections are allowed. Required parameter: `<hosts>` New hosts from which connections are allowed.
`-getO[neACLDir]`	Get the `oneacldir`.
`-setO[neACLDir]`	Set the `oneacldir`. Required parameter: `<directory>` New ACL directory.
`-getPo[rt]`	Get the current Administration Server port number.
`-setPo[rt]`	Set the Administration Server port number. Required parameter: `<port>` New server port number.
`-getSe[rverAddress]`	Get the current Administration Server address.
`-setSe[rverAddress]`	Set the Administration Server address. Required parameter: `<address>` New server address.
`-getSu[iteSpotUser]`	Get the user name that the server is currently running as.
`-setSu[iteSpotUser]`	Set the user name that the server should run as. Required parameter include: `<user>` New user name that the server should run as.
`-r[estart]`	Restart the Administration Server.
`-st[op]`	Stop the Administration Server.

admin_ip.pl

This tool is useful when, for any reason, your computer system's IP address changes. The IP address must be changed in both the Administration Server configuration and the Configuration Directory, or you won't be able to start the Administration Server.

This Perl script automatically changes the Administration Server IP address in both the `local.conf` file and in the Configuration Directory. The old IP address is stored in the file `local.conf.old`. The `admin_ip.pl` file is stored in the `<Server_Root>\shared\bin` directory.

To run `admin_ip.pl`:

In the `<Server_Root>\shared\bin directory,`

- On Windows NT, enter:

```
..\..\install\perl admin_ip.pl <Directory_Manager_DN>
<Directory_Manager_password> <old_IP> <new_IP> [port #]
```

- On Unix, enter:

```
admin_ip.pl <Directory_Manager_DN> <Directory_Manager_password>
<old_IP> <new_IP> [port #]
```

ldapsearch and ldapmodify

These are tools for searching and modifying the user directory. They are stored at `<server_root>/shared/bin`. For detailed information, see the *Directory Server Administrator's Guide.*

sec-migrate

The `sec-migrate` command migrates keys and certificates from a pre-4.0 Netscape server to a target Netscape 4.0 server. This is useful when want to use a pre-4.0 SSL certificate with a new 4.x server. Using this command allows you to use the existing pre-4.0 certificate instead of obtaining a new one. `Sec-migrate` is stored at `<server_root>/bin/admin/admin/bin`.

Syntax

`sec-migrate` *[src] [alias] [dist] [sie] [passwd]*

Enter information for the following variables:

src. Pre-4.0 server root.

alias. Alias of the old key database.

dist. Target 4.0 server root.

sie. Instance name of 4.0 server.

passwd. Password used to generate pre-4.0 key database.

modutil

The Security Module Database Tool is a command-line utility for managing PKCS #11 module information within `secmod.db` files or within hardware tokens. You can use the tool to add and delete PKCS #11 modules, change passwords, set defaults, list module contents, enable or disable slots, enable or disable FIPS-140-1 compliance, and assign default providers for cryptographic operations. This tool can also create `key3.db`, `cert7.db`, and `secmod.db` security database files.

The tasks associated with security module database management are part of a process that typically also involves managing key databases (`key3.db` files) and certificate databases (`cert7.db` files). The key, certificate, and PKCS #11 module management process generally begins with creating the keys and key database necessary to generate and manage certificates and the certificate database.

This tool is available for Solaris 2.5.1 (SunOS 5.5.1) and Windows NT 4.0. It is stored at `<server_root>/shared/bin`.

Syntax

To run the Security Module Database Tool, type the command modutil option *[arguments]*

where `option` and [*`arguments]`*] are combinations of the options and arguments listed in the following section. Each command takes one option. Each option may take zero or more arguments. To see a usage string, issue the command without options.

Options and Arguments

Options specify an action. Option arguments modify an action. The options and arguments for the `modutil` command are defined as follows:

Table B.3 Options and Arguments for modutil

Options	Description
-create	Create new `secmod.db`, `key3.db`, and `cert7.db` files. Use the `-dbdir` *directory* argument to specify a directory. If any of these databases already exist in a specified directory, the Security Module Database Tool displays an error message.
-list [modulename]	Display basic information about the contents of the `secmod.db` file. Use *modulename* to display detailed information about a particular module and its slots and tokens.
-add modulename	Add the named PKCS #11 module to the database. Use this option with the `-libfile`, `-ciphers`, and `-mechanisms` arguments.

Table B.3 Options and Arguments for modutil

-jar JAR-file	Add a new PKCS #11 module to the database using the named JAR file. For more information, see “JAR Installation File” on page 169. Use this option with the `-installdir` and `-tempdir` arguments. The JAR file uses the Netscape Server PKCS #11 JAR format to identify all the files to be installed, the module’s name, the mechanism flags, and the cipher flags. The JAR file should also contain any files to be installed on the target machine, including the PKCS #11 module library file and other files such as documentation. See the section JAR Installation File for information on creating the special script needed to perform an installation through a server or with the Security Module Database Tool (that is, in environments without JavaScript support).
-delete modulename	Delete the named module. Note that you cannot delete the Netscape Communicator internal PKCS #11 module.
-changepw tokenname	Change the password on the named token. If the token has not been initialized, this option initializes the password. Use this option with the `-pwfile` and `-newpwfile` arguments. In this context, the term “password” is equivalent to a personal identification number (PIN).
-default modulename	Specify the security mechanisms for which the named module will be a default provider. The security mechanisms are specified with the `-mechanisms` *mechanism-list* argument.
-undefault modulename	Specify the security mechanisms for which the named module will *not* be a default provider. The security mechanisms are specified with the `-mechanisms` *mechanism-list* argument.
-enable modulename	Enable all slots on the named module. Use the `[-slot` *slotname*`]` argument to enable a specific slot.
-disable modulename	Disable all slots on the named module. Use the `[-slot` *slotname*`]` argument to disable a specific slot.
-fips [true | false]	Enable (`true`) or disable (`false`) FIPS-140-1 compliance for the Netscape Communicator internal module.

Table B.3 Options and Arguments for modutil

-force	Disable the Security Module Database Tool's interactive prompts so it can be run from a script. Use this option only after manually testing each planned operation to check for warnings and to ensure that bypassing the prompts will cause no security lapses or loss of database integrity.
Arguments	**Description**
-dbdir directory	Specify a directory in which to access or create security module database files. On Unix, the Security Module Database Tool defaults to the user's Netscape directory. Windows NT has no default directory, so `-dbdir` must be used to specify a directory.
-libfile library-file	Specify a path to the DLL or other library file containing the implementation of the PKCS #11 interface module that is being added to the database.
-ciphers cipher-enable-list	Enable specific ciphers in a module that is being added to the database. The *cipher-enable-list* is a colon-delimited list of cipher names. Enclose this list in quotation marks if it contains spaces. The following cipher is currently available: `FORTEZZA`.
-mechanisms mechanism-list	Specify the security mechanisms for which a particular module will be flagged as a default provider. The *mechanism-list* is a colon-delimited list of mechanism names. Enclose this list in quotation marks if it contains spaces. The module becomes a default provider for the listed mechanisms when those mechanisms are enabled. If more than one module claims to be a particular mechanism's default provider, that mechanism's default provider is undefined. The following mechanisms are currently available: `RSA`, `DSA`, `RC2`, `RC4`, `RC5`, `DES`, `DH`, `FORTEZZA`, `SHA1`, `MD5`, `MD2`, `RANDOM` (for random number generation), and `FRIENDLY` (meaning certificates are publicly readable).
-installdir *root-installation-directory*	Specify the root installation directory relative to which files will be installed by the `-jar` *JAR-file* option. This directory should be one below which it is appropriate to store dynamic library files (for example, a server's root directory or the Netscape Communicator root directory).

Table B.3 Options and Arguments for modutil

-tempdir temporary-directory	The temporary directory is the location where temporary files will be created in the course of installation by the `-jar` *JAR-file* option. If no temporary directory is specified, the current directory will be used.
-pwfile old-password-file	Specify a text file containing a token's existing password so that a password can be entered automatically when the `-changepw` *tokenname* option is used to change passwords.
-newpwfile new-password-file	Specify a text file containing a token's new or replacement password so that a password can be entered automatically with the `-changepw` *tokenname* option.
-slot slotname	Specify a particular slot to be enabled or disabled with the `-enable` *modulename* or `-disable` *modulename* options.
-nocertdb	Do not open the certificate or key databases. This has several effects: • With the `-create` command, only a `secmod.db` file will be created; `cert7.db` and `key3.db` will not be created. • With the `-jar` command, signatures on the JAR file will not be checked. • With the `-changepw` command, the password on the Netscape internal module cannot be set or changed, since this password is stored in `key3.db`.

Usage

The Security Module Database Tool's capabilities are grouped as follows, using these combinations of options and arguments. The options and arguments in square brackets are optional, those without square brackets are required.

- Creating a set of security management database files (`key3.db`, `cert7.db`, and `secmod.db`):

  ```
  -create
  ```

- Displaying basic module information or detailed information about the contents of a given module:

  `-list` [modulename]

- Adding a PKCS #11 module, which includes setting a supporting library file, enabling ciphers, and setting default provider status for various security mechanisms:

  `-add` modulename `-libfile` library-file `[-ciphers` cipher-enable-list`]` `[-mechanisms` mechanism-list`]`

- Adding a PKCS #11 module from an existing JAR file:

  `-jar` JAR-file `-installdir` root-installation-directory
  `[-tempdir` temporary-directory`]`

- Deleting a specific PKCS #11 module from a security module database:

  `-delete` modulename

- Initializing or changing a token's password:

  `-changepw` tokenname [-pwfile old-password-file]
  [-newpwfile new-password-file]

- Setting the default provider status of various security mechanisms in an existing PKCS #11 module:

  `-default` modulename -mechanisms mechanism-list

- Clearing the default provider status of various security mechanisms in an existing PKCS #11 module:

  `-undefault` modulename -mechanisms mechanism-list

- Enabling a specific slot or all slots within a module:

  -enable modulename [-slot slotname]

- Disabling a specific slot or all slots within a module:

  -disable modulename [-slot slotname]

- Enabling or disabling FIPS-140-1 compliance within the Netscape Communicator internal module:

```
-fips [true | false]
```

- Disabling interactive prompts for the Security Module Database Tool, to support scripted operation:

```
-force
```

JAR Installation File

When a JAR file is run by a server, by the Security Module Database Tool, or by any program that does not interpret JavaScript, a special information file must be included in the format described below.

This information file contains special scripting and must be declared in the JAR archive's manifest file. The script can have any name. The metainfo tag for this is `Pkcs11_install_script`. To declare meta-information in the manifest file, put it in a file that is passed to the Netscape Signing Tool.

Sample Script

For example, the PKCS #11 installer script could be in the file `pk11install.` If so, the metainfo file for the Netscape Signing Tool would include a line such as this:

```
+ Pkcs11_install_script: pk11install
```

The sample script file could contain the following:

```
ForwardCompatible { IRIX:6.2:mips SUNOS:5.5.1:sparc }
Platforms {
   WINNT::x86 {
      ModuleName { "Fortezza Module" }
      ModuleFile { win32/fort32.dll }
      DefaultMechanismFlags{0x0001}
      DefaultCipherFlags{0x0001}
      Files {
         win32/setup.exe {
            Executable
            RelativePath { %temp%/setup.exe }
         }
         win32/setup.hlp {
            RelativePath { %temp%/setup.hlp }
         }
         win32/setup.cab {
            RelativePath { %temp%/setup.cab }
         }
      }
   }
   WIN95::x86 {
      EquivalentPlatform {WINNT::x86}
   }
   SUNOS:5.5.1:sparc {
      ModuleName { "Fortezza UNIX Module" }
      ModuleFile { unix/fort.so }
      DefaultMechanismFlags{0x0001}
      CipherEnableFlags{0x0001}
      Files {
         unix/fort.so {
            RelativePath{%root%/lib/fort.so}
            AbsolutePath{/usr/local/netscape/lib/fort.so}
            FilePermissions{555}
         }
         xplat/instr.html {
            RelativePath{%root%/docs/inst.html}
            AbsolutePath{/usr/local/netscape/docs/inst.html}
            FilePermissions{555}
         }
      }
   }
   IRIX:6.2:mips {
      EquivalentPlatform { SUNOS:5.5.1:sparc }
   }
}
```

Script Grammar

The script file grammar is as follows:

```
--> valuelist

valuelist --> value valuelist
               <null>

value ---> key_value_pair
            string

key_value_pair --> key { valuelist }

key --> string

string --> simple_string
            "complex_string"
```

`simple_string` `--> [^ \t\n\""{""}"]+` (No whitespace, quotes, or braces.)

`complex_string` `--> ([^\"\\\r\n]|(\\\")|(\\\\))+` (Quotes and

backslashes must be escaped with a backslash. A complex string must not include newlines or carriage returns.)

Outside of complex strings, all white space (for example, spaces, tabs, and carriage returns) is considered equal and is used only to delimit tokens.

Keys

Keys are case-insensitive. This section discusses the following keys:

Global Keys
Per-Platform Keys
Per-File Keys

Global Keys

ForwardCompatible

Gives a list of platforms that are forward compatible. If the current platform cannot be found in the list of supported platforms, then the `ForwardCompatible` list is checked for any platforms that have the same OS and architecture in an earlier version. If one is found, its attributes are used for the current platform.

`Platforms` (required)

Gives a list of platforms. Each entry in the list is itself a key-value pair: the key is the name of the platform and the value list contains various attributes of the platform. The `ModuleName`, `ModuleFile`, and `Files` attributes must be specified for each platform unless an `EquivalentPlatform` attribute is specified. The platform string is in the following format: *system name*:*OS release*:*architecture*. The installer obtains these values from NSPR. *OS release* is an empty string on non-Unix operating systems. The following system names and platforms are currently defined by NSPR:

- AIX (rs6000)
- BSDI (x86)
- FREEBSD (x86)
- HPUX (hppa1.1)

- IRIX (mips)
- LINUX (ppc, alpha, x86)
- MacOS (PowerPC)
- NCR (x86)
- NEC (mips)
- OS2 (x86)
- OSF (alpha)
- ReliantUNIX (mips)
- SCO (x86)
- SOLARIS (sparc)
- SONY (mips)
- SUNOS (sparc)
- UnixWare (x86)
- WIN16 (x86)
- WIN95 (x86)
- WINNT (x86)

Here are some examples of valid platform strings:

```
IRIX:6.2:mips
SUNOS:5.5.1:sparc
Linux:2.0.32:x86
WIN95::x86.
```

Per-Platform Keys

These keys have meaning only within the value list of an entry in the `Platforms` list.

`ModuleName` (required)

Gives the common name for the module. This name will be used to reference the module from Netscape Communicator, the Security Module Database tool (*modutil*), servers, or any other program that uses the Netscape security module database.

ModuleFile (required)

Names the PKCS #11 module file (DLL or `.so`) for this platform. The name is given as the relative path of the file within the JAR archive.

Files (required)

Lists the files that need to be installed for this module. Each entry in the file list is a key-value pair: the key is the path of the file in the JAR archive, and the value list contains attributes of the file. At least `RelativePath` or `AbsolutePath` must be specified for each file.

DefaultMechanismFlags

Specifies mechanisms for which this module will be a default provider. This key-value pair is a bitstring specified in hexadecimal (0x) format. It is constructed as a bitwise OR of the following constants. If the `DefaultMechanismFlags` entry is omitted, the value defaults to 0x0.

RSA:	0x00000001
DSA:	0x00000002
RC2:	0x00000004
RC4:	0x00000008
DES:	0x00000010
DH:	0x00000020
FORTEZZA:	0x00000040
RC5:	0x00000080
SHA1:	0x00000100
MD5:	0x00000200
MD2:	0x00000400
RANDOM:	0x08000000
FRIENDLY:	0x10000000
OWN_PW_DEFAULTS:	0x20000000
DISABLE:	0x40000000

CipherEnableFlags

Specifies ciphers that this module provides but Netscape products do not, so that Netscape products can enable them. This key is a bitstring specified in hexadecimal (0x) format. It is constructed as a bitwise OR of the following constants. If the `CipherEnableFlags` entry is omitted, the value defaults to 0x0.

```
    FORTEZZA:              0x0000 0001
```

EquivalentPlatform

Specifies that the attributes of the named platform should also be used for the current platform. Saves typing when there is more than one platform using the same settings.

Per-File Keys

These keys have meaning only within the value list of an entry in a `Files` list. At least one of `RelativePath` and `AbsolutePath` must be specified. If both are specified, the relative path is tried first, and the absolute path is used only if no relative root directory is provided by the installer program.

`RelativePath`

Specifies the destination directory of the file, relative to some directory decided at install time. Two variables can be used in the relative path: "`%root%`" and "`%temp%`". "`%root%`" is replaced at run time with the directory relative to which files should be installed; for example, it may be the server's root directory or the Netscape Communicator root directory. The "`%temp%`" directory is created at the beginning of the installation and destroyed at the end.

The purpose of "`%temp%`" is to hold executable files (such as setup programs) or files that are used by these programs. For example, a Windows installation might consist of a `setup.exe` installation program, a help file, and a `.cab` file containing compressed information. All these files could be installed in the temporary directory. Files destined for the temporary directory are guaranteed to be in place before any executable file is run; they are not deleted until all executable files have finished.

`AbsolutePath`

Specifies the destination directory of the file as an absolute path. If both `RelativePath` and `AbsolutePath` are specified, the installer attempts to use the relative path; if it is unable to determine a relative path, it uses the absolute path.

`Executable`

Specifies that the file is to be executed during the course of the installation. Typically this string would be used for a setup program provided by a module vendor, such as a self-extracting `setup.exe`. More than one file can be specified as executable, in which case the files are run in the order in which they are specified in the script file.

FilePermissions

Interpreted as a string of octal digits, according to the standard Unix format. This string is a bitwise OR of the following constants:

```
user read:                0400
user write:               0200
user execute:             0100
group read:               0040
group write:              0020
group execute:            0010
other read:               0004
other write:              0002
other execute:       0001
```

Some platforms may not understand these permissions. They are applied only insofar as they make sense for the current platform. If this attribute is omitted, a default of 777 is assumed.

Examples

Creating Database Files
Displaying Module Information
Setting a Default Provider
Enabling a Slot
Enabling FIPS Compliance
Adding a Cryptographic Module
Installing a Cryptographic Module from a JAR File
Changing the Password on a Token

Creating Database Files

This example creates a set of security management database files in the specified directory:

```
modutil -create -dbdir c:\databases
```

The Security Module Database Tool displays a warning:

```
WARNING: Performing this operation while a Netscape product is running
could cause corruption of your security databases. If a Netscape product
is currently running, you should exit the product before continuing this
operation. Type ’q <enter>’ to abort, or <enter> to continue:
```

After you press Enter, the tool displays the following:

```
Creating "c:\databases\key3.db"...done.
Creating "c:\databases\cert7.db"...done.
Creating "c:\databases\secmod.db"...done.
```

Displaying Module Information

This example gives detailed information about the specified module:

```
modutil -list "Netscape Internal PKCS #11 Module" -dbdir c:\databases
```

The Security Module Database Tool displays information similar to this:

```
Using database directory c:\databases...
--------------------------------------------------------
Name: Netscape Internal PKCS #11 Module
Library file: **Internal ONLY module**
Manufacturer: Netscape Communications Corp.
Description: Communicator Internal Crypto Svc
PKCS #11 Version 2.0
Library Version: 4.0
Cipher Enable Flags: None
Default Mechanism Flags: RSA:DSA:RC2:RC4:DES:SHA1:MD5:MD2

Slot: Communicator Internal Cryptographic Services Version 4.0
Manufacturer: Netscape Communications Corp.
Type: Software
Version Number: 4.1
Firmware Version: 0.0
Status: Enabled
Token Name: Communicator Generic Crypto Svcs
Token Manufacturer: Netscape Communications Corp.
Token Model: Libsec 4.0
Token Serial Number: 0000000000000000
Token Version: 4.0
Token Firmware Version: 0.0
Access: Write Protected
Login Type: Public (no login required)
User Pin: NOT Initialized

Slot: Communicator User Private Key and Certificate Services
Manufacturer: Netscape Communications Corp.
Type: Software
```

```
Version Number: 3.0
Firmware Version: 0.0
Status: Enabled
Token Name: Communicator Certificate DB
Token Manufacturer: Netscape Communications Corp.
Token Model: Libsec 4.0
Token Serial Number: 0000000000000000
Token Version: 7.0
Token Firmware Version: 0.0
Access: NOT Write Protected
Login Type: Login required
User Pin: NOT Initialized
```

Setting a Default Provider

This example makes the specified module a default provider for the RSA, DSA, and RC2 security mechanisms:

```
modutil -default "Cryptographic Module" -dbdir c:\databases -mechanisms
RSA:DSA:RC2
```

The Security Module Database Tool displays a warning:

```
WARNING: Performing this operation while a Netscape product is running
could cause corruption of your security databases. If a Netscape product
is currently running, you should exit the product before continuing this
operation. Type ’q <enter>’ to abort, or <enter> to continue:
```

After you press Enter, the tool displays the following:

```
Using database directory c:\databases...

Successfully changed defaults.
```

Enabling a Slot

This example enables a particular slot in the specified module:

```
modutil -enable "Cryptographic Module" -slot "Cryptographic Reader" -
dbdir c:\databases
```

The Security Module Database Tool displays a warning:

```
WARNING: Performing this operation while a Netscape product is running
could cause corruption of your security databases. If a Netscape product
is currently running, you should exit the product before continuing this
operation. Type ’q <enter>’ to abort, or <enter> to continue:
```

After you press Enter, the tool displays the following:

```
Using database directory c:\databases...
```

```
Slot "Cryptographic Reader" enabled.
```

Enabling FIPS Compliance

This example enables FIPS-140-1 compliance in Communicator’s internal module:

```
modutil -fips true
```

The Security Module Database Tool displays a warning:

```
WARNING: Performing this operation while a Netscape product is running
could cause corruption of your security databases. If a Netscape product
is currently running, you should exit the product before continuing this
operation. Type ’q <enter>’ to abort, or <enter> to continue:
```

After you press Enter, the tool displays the following:

```
FIPS mode enabled.
```

Adding a Cryptographic Module

This example adds a new cryptographic module to the database:

```
C:\modutil> modutil -dbdir "C:\databases" -add "Cryptorific Module" -
libfile "C:\winnt\system32\crypto.dll" -mechanisms RSA:DSA:RC2:RANDOM
```

The Security Module Database Tool displays a warning:

```
WARNING: Performing this operation while a Netscape product is running
could cause corruption of your security databases. If a Netscape product
is currently running, you should exit the product before continuing this
operation. Type ’q <enter>’ to abort, or <enter> to continue:
```

After you press Enter, the tool displays the following:

```
Using database directory C:\databases...
Module "Cryptorific Module" added to database.
C:\modutil>
```

Installing a Cryptographic Module from a JAR File

This example installs a cryptographic module from the following sample installation script.

```
Platforms {
   WinNT::x86 {
      ModuleName { "Cryptorific Module" }
      ModuleFile { crypto.dll }
      DefaultMechanismFlags{0x0000}
      CipherEnableFlags{0x0000}
      Files {
         crypto.dll {
            RelativePath{ %root%/system32/crypto.dll }
         }
         setup.exe {
            Executable
            RelativePath{ %temp%/setup.exe }
         }
      }
   }
   Win95::x86 {
      EquivalentPlatform { Winnt::x86 }
   }
}
```

To install from the script, use the following command. The root directory should be the Windows root directory (for example, `c:\\windows`, or `c:\\winnt`).

```
C:\modutil> modutil -dbdir "c:\databases" -jar install.jar -installdir
"C:/winnt"
```

The Security Module Database Tool displays a warning:

```
WARNING: Performing this operation while a Netscape product is running
could cause corruption of your security databases. If a Netscape product
is currently running, you should exit the product before continuing this
operation. Type ’q <enter>’ to abort, or <enter> to continue:
```

After you press Enter, the tool displays the following:

```
Using database directory c:\databases...

This installation JAR file was signed by:
----------------------------------------------

**SUBJECT NAME**

C=US, ST=California, L=Mountain View, CN=Cryptorific Inc., OU=Digital ID
Class 3 - Netscape Object Signing, OU="www.verisign.com/repository/CPS
Incorp. by Ref.,LIAB.LTD(c)9 6", OU=www.verisign.com/CPS Incorp.by Ref.
LIABILITY LTD.(c)97 VeriSign, OU=VeriSign Object Signing CA - Class 3
Organization, OU="VeriSign, Inc.", O=VeriSign Trust Network **ISSUER
NAME**, OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97
VeriSign, OU=VeriSign Object Signing CA - Class 3 Organization,
OU="VeriSign, Inc.", O=VeriSign Trust Network
----------------------------------------------
```

```
Do you wish to continue this installation? (y/n) y
Using installer script "installer_script"
Successfully parsed installation script
Current platform is WINNT::x86
Using installation parameters for platform WinNT::x86
Installed file crypto.dll to C:/winnt/system32/crypto.dll
Installed file setup.exe to ./pk11inst.dir/setup.exe
Executing "./pk11inst.dir/setup.exe"...
"./pk11inst.dir/setup.exe" executed successfully
Installed module "Cryptorific Module" into module database

Installation completed successfully
C:\modutil>
```

Changing the Password on a Token

This example changes the password for a token on an existing module.

```
C:\modutil> modutil -dbdir "c:\databases" -changepw "Communicator
Certificate DB"
```

The Security Module Database Tool displays a warning:

```
WARNING: Performing this operation while a Netscape product is running
could cause corruption of your security databases. If a Netscape product
is currently running, you should exit the product before continuing this
operation. Type ’q <enter>’ to abort, or <enter> to continue:
```

After you press Enter, the tool displays the following:

```
Using database directory c:\databases...
Enter old password:
Incorrect password, try again...
Enter old password:
Enter new password:
Re-enter new password:
Token "Communicator Certificate DB" password changed successfully.
C:\modutil>
```

Appendix

C

FORTEZZA

FORTEZZA is a cryptographic system that combines the use of hardware-based tokens and software-based algorithms to secure web-based information exchange. The US government developed FORTEZZA to manage sensitive but unclassified information.

How It Works

FORTEZZA provides a higher level of security than typical encryption systems because it requires three elements:

- A Crypto Card
- FORTEZZA encryption algorithms
- FORTEZZA key management.

First, the US government provides your department or agency access to a Certificate Authority Workstation (CAW). The workstation itself may or may not be located at your worksite. A Certificate Authority (CA) representing your department or agency operates the CAW. The CA may be a security office or other designee who establishes, authenticates, and programs FORTEZZA Crypto Cards. A FORTEZZA Crypto Card is a PCMCIA card that has been activated and issued by the CA. The CA also maintains and revokes user keys and certificates as necessary.

Information System (IS) administrators install FORTEZZA software and card readers on some or all of your enterprise servers, and then card readers are installed on your users' computers or workstations. Netscape FORTEZZA products are designed to operate properly with any PCMCIA-compliant card reader that is supported by the Litronic device driver.

Each enterprise user must request and obtain a FORTEZZA Crypto Card from a CA.

Typically, a user who wants to access a FORTEZZA-secured server plugs the FORTEZZA Crypto Card into the PCMCIA reader. By inserting the card and typing in the Personal Identification Number (PIN), the user tells the client to

- load all of the CA certificates on the card into memory
- trust the CA certificates provided on the card
- if requested, use the keys on the card for client authentication

How FORTEZZA Crypto Cards are Certified

The US government established the Policy Approval Authority (PAA), a regulating body, to ensure that only valid users are given authenticated FORTEZZA cards.

The PAA delegates its authority to Policy Creation Authorities (PCAs). These are groups that may represent a branch of the government or a large corporation. PCAs in turn delegate authority to Certification Authorities (CAs).

Certification Authorities are the individual who actually verify users' key information. CAs program, activate, and issue cards to government employees and to individuals who conduct business with the government. A single CA might handle the encryption needs of a small company, a single department in a large company, or a department in a government agency.

FORTEZZA Keys, Certificates, and Encryption

The CA programs FORTEZZA Crypto Cards with any combination of encryption and key management approaches. Some of these are described briefly here. For more information about how keys, certificates, and encryption work in general, see Appendix D, “Introduction to Public-Key Cryptography,” and Appendix E, “Introduction to SSL,” in this manual.

Encryption Algorithms

SKIPJACK. Data encryption and decryption algorithms typically used with the SSL protocol.

SSL Protocol. Symmetric encryption nested within public-key encryption and authenticated through the use of certificates.

RC4 Encryption. A kind of 128-bit software encryption. Servers use this kind of encryption to optimize performance.

NULL encryption. Typically used when providing only access control or when using pre-encrypted fields.

Key Management

Certificate revocation list (CRL). A list, provided by the CA, of all revoked certificates.

Compromised key list (CKL). A list of key information about users who have compromised keys. The CA also provides this list.

Enabling FORTEZZA

To set up FORTEZZA, use the Certificate Setup Wizard as described in “Obtaining and Installing a Certificate” on page 67. Be sure to indicate FORTEZZA when appropriate:

- When prompted to request or install a certificate, be sure to indicate that you’re using FORTEZZA. Because your certificate information comes from the Crypto Card manufacturer with the certificate installed, you can bypass those steps in the wizard.
- When prompted to choose ciphers, be sure to choose FORTEZZA ciphers.

If you’re going to use both internal and external SSL tokens, use the Certificate Setup Wizard two times. During the first use, select the Internal token. During the second use, indicate the External (FORTEZZA) token.

Note Each Netscape server that supports FORTEZZA may have its own setup options and requirements. See the *Administrator’s Guide* for your server for related information.

Appendix

D

Introduction to Public-Key Cryptography

Public-key cryptography and related standards and techniques underlie security features of many Netscape products, including signed and encrypted email, form signing, object signing, single sign-on, and the Secure Sockets Layer (SSL) protocol. This document introduces the basic concepts of public-key cryptography.

- Internet Security Issues
- Encryption and Decryption
- Digital Signatures
- Certificates and Authentication
- Managing Certificates

For more information on these topics and other aspects of cryptography, see Security Resources.

For an overview of SSL, see 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.** A person can pretend to be someone else. For example, a person can pretend to have the email address `jdoe@mozilla.com`, or a computer can identify itself as a site called `www.mozilla.com` when it is not. This type of impersonation is known as spoofing.
    - **Misrepresentation.** A person or organization can misrepresent 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 will be 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.

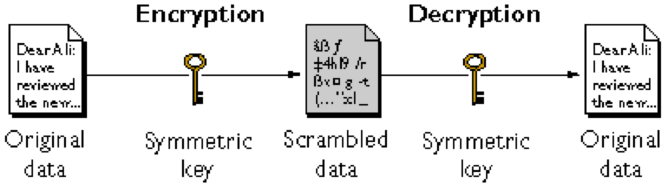
- Symmetric-Key Encryption

- 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, as shown in Figure D.1.

Figure D.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.

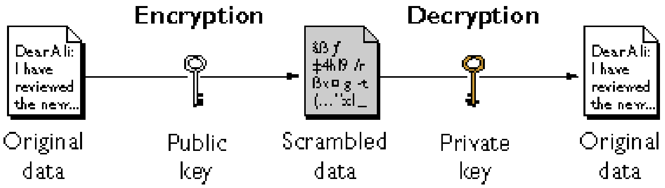
Symmetric-key encryption plays an important role in the SSL protocol, which is widely used for authentication, tamper detection, and encryption over TCP/IP networks. SSL also uses techniques of public-key encryption, which is described in the next section.

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.) Data encrypted with your public key can be decrypted only with your private key. Figure D.2 shows a simplified view of the way public-key encryption works.

Figure D.2 Public-key encryption



The scheme shown in Figure D.2 lets you freely distribute a public key, and only you will be able to 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 D.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 such as Communicator 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 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 prime 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 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.

Because the ability to surreptitiously intercept and decrypt encrypted information has historically been a significant military asset, the U.S. Government restricts export of cryptographic software, including most software that permits use of symmetric encryption keys longer than 40 bits. For detailed information about these restrictions as they apply to Netscape products, see Export Restrictions on International Sales.

Digital Signatures

Encryption and decryption address the problem of eavesdropping, one of the three Internet security issues mentioned at the beginning of this document. But encryption and decryption, by themselves, do not address the other two problems mentioned in Internet Security Issues: 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, 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 D.3 shows a simplified view of the way a digital signature can be used to validate the integrity of signed data.

Figure D.3 Using a digital signature to validate data integrity

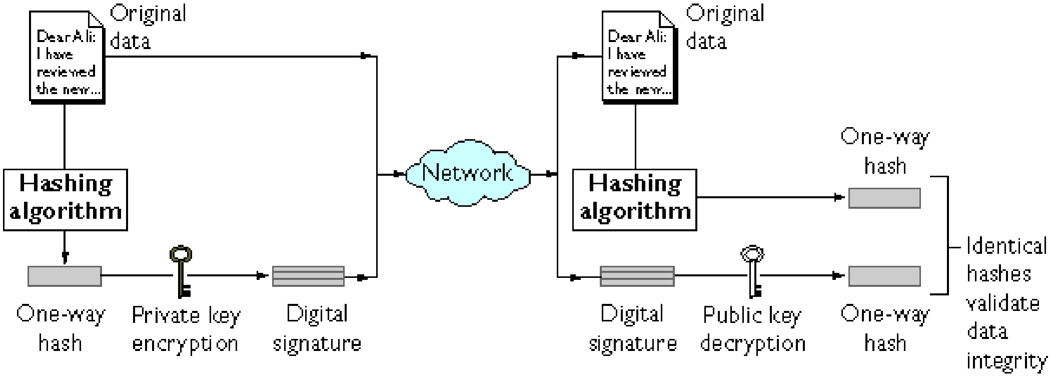


Figure D.3 shows two items transferred to the recipient of some signed data: the original data and the digital signature, which is basically a one-way hash (of the original data) that has been encrypted with the signer’s private key. To validate the integrity of the data, the receiving software first 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 Certificates and Authentication.

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 and to associate 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).

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 Netscape Certificate Server). 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 will be 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 will work 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.

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:

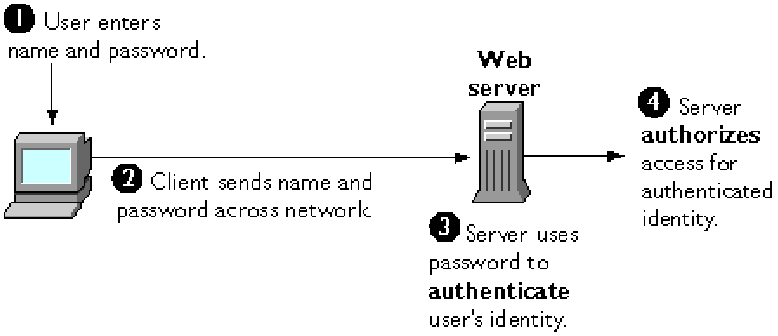
- Password-Based Authentication. Almost all server software permits client authentication by means of a name and password. For example, a server might require a user to type a name and password before granting access to the server. The server maintains a list of names and passwords; if a particular name 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 D.4 shows the basic steps involved in authenticating a client by means of a name and password. Figure 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 D.4 Using a password to authenticate a client to a server


1 User enters name and password.
Web server
2 Client sends name and password across network.
3 Server uses password to **authenticate** user's identity.
4 Server **authorizes** access for authenticated identity.

These are the steps shown in Figure 2:

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.

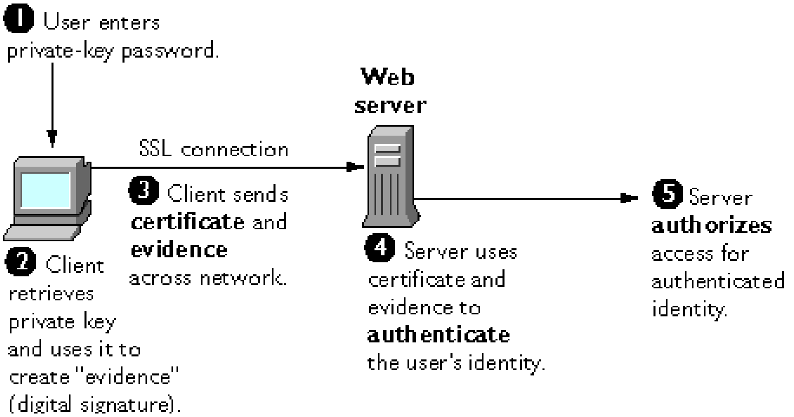
As shown in the next section, one of the advantages of certificate-based authentication is that it can be used to replace the first three steps in Figure 2 with a mechanism that allows the user to supply just one password (which is not sent across the network) and allows the administrator to control user authentication centrally.

Certificate-Based Authentication

Figure D.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 D.4, Figure D.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 D.5 Using a certificate to authenticate a client to a server



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 D.4, the process shown in Figure D.5 requires the use of SSL. Figure D.5 also assumes that the client has a valid certificate that can be used to identify the client to the server. Certficate-based authentication is generally considered prefereable to password-based atuthentication because it is based on wheat 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 unuthorized 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 reasonable frequeent 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 sightn some data corresponds to the public key in a certificate. It is the user's responsiblity to protect a machine's physcial security and to keep the private-key password secret.

These are the steps shown in Figure D.3:

1. The client software, such as Communicator, 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 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 D.5 to Figure D.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 Netscape 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 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. See SSL Protocol.

  Example: Internet sites that engage in electronic commerce (commonly known as **e-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 email. 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. 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. See Object Signing.

  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. See How CA Certificates Are Used to Establish Trust.

  Example: The CA certificates stored in Communicator determine what other certificates that copy of Communicator can authenticate. An administrator can implement some aspects of corporate security policies by controlling the CA certificates stored in each user’s copy of Communicator.

The sections that follow describes how certificates are used by Netscape products.

SSL Protocol

The Secure Sockets Layer (SSL) protocol, which was originally developed by Netscape, 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. For more detailed information about SSL, see Introduction to SSL.

Signed and Encrypted Email

Some email programs (including Messenger, which is part of Communicator) support digitally signed and encrypted email using a widely accepted protocol known as Secure Multipurpose Internet Mail Extension (S/MIME). Using S/MIME to sign or encrypt email messages requires the sender of the message to have an S/MIME certificate.

An email 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 email software on the receiving end, the user will be 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 email also provides some assurance that the email has not been tampered with. As discussed at the beginning of this document, this kind of assurance is known as nonrepudiation. In other words, signed email 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.)

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.

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 a different password to log into the network, collect email, 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 Netscape products relies on SSL client authentication (see Certificate-Based Authentication). 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.

For information about the single sign-on support currently provided by Netscape products, see Single Sign-On Deployment Guide.

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 email, but e-commerce often involves filling in a form on a web page rather than sending an email.

The Netscape 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 the client SSL and S/MIME certificates that are installed in Communicator. 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 Netscape utility called the Signature Verification Tool to validate the digital signature.

For more information about support for form signing in Netscape products, see Netscape Form Signing.

Object Signing

Communicator and other Netscape 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.

For more information about support for object signing in Netscape products, see Netscape Object Signing: Establishing Trust for Downloaded Software.

Contents of a Certificate

The contents of certificates supported by Netscape 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 Netscape Communications Corp.:

```
uid=doe,e=doe@netscape.com,cn=John Doe,o=Netscape Communications
Corp.,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 document. For comprehensive information about DNs, see A String Representation of Distinguished Names.

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
  - 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, 1996 and 1:00 p.m. November 15, 1997)

  - 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 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=Ace Industry, C=US
    Validity:
      Not Before: Fri Oct 17 18:36:25 1997
      Not  After: Sun Oct 17 18:36:25 1999
    Subject: CN=Jane Doe, OU=Finance, O=Ace 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 the same 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 Netscape Certificate Server). A list of third-party certificate authorities is available at Certificate Authority Services.

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.

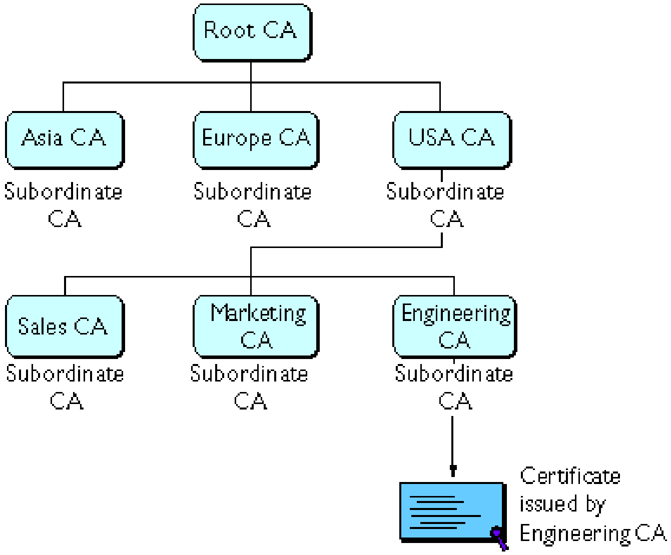
- CA Hierarchies
- Certificate Chains
- Verifying a Certificate Chain

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 like that shown in Figure D.6.

Figure D.6 Example of a hierarchy of certificate authorities



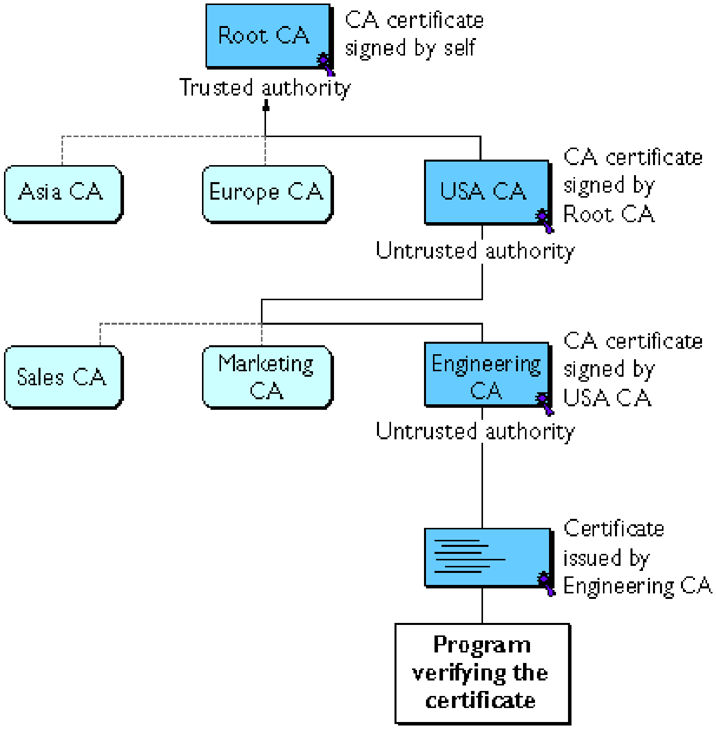
In this model, the root CA is at the top of the hierarchy. The root CA's certificate is a **self-signed certificate:** that is, the certificate is digitally signed by the same entity--the root CA--that the certificate identifies. The CAs that are directly subordinate to the root CA have CA certificates 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 D.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 D.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 D.6).

Figure D.7 Example of 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 D.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 D.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. Netscape 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 Netscape certificate type extension, and chain verification returns to step 1 to start again, but with this new certificate. Figure D.8 presents an example of this process.

Figure D.8 Verifying a certificate chain all the way to the root CA

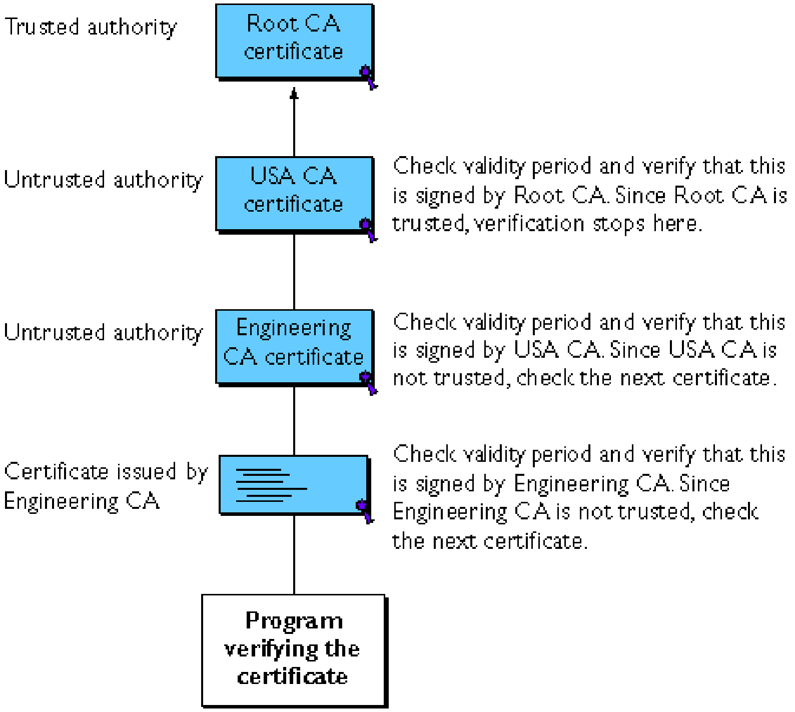
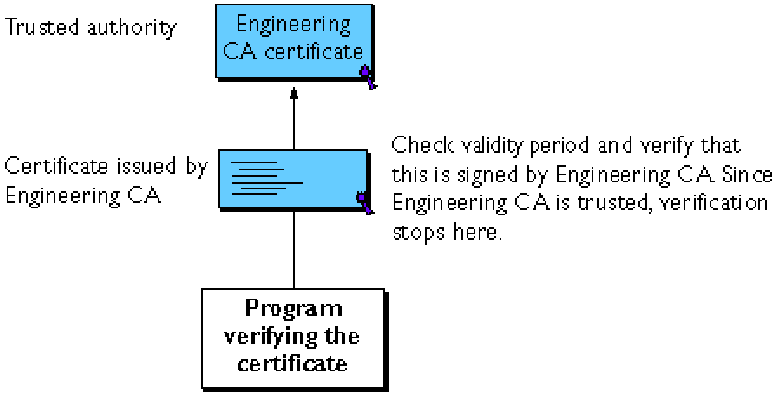

Trusted authority
Root CA certificate
Untrusted authority
USA CA certificate
Check validity period and verify that this is signed by Root CA. Since Root CA is trusted, verification stops here.
Untrusted authority
Engineering CA certificate
Check validity period and verify that this is signed by USA CA. 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

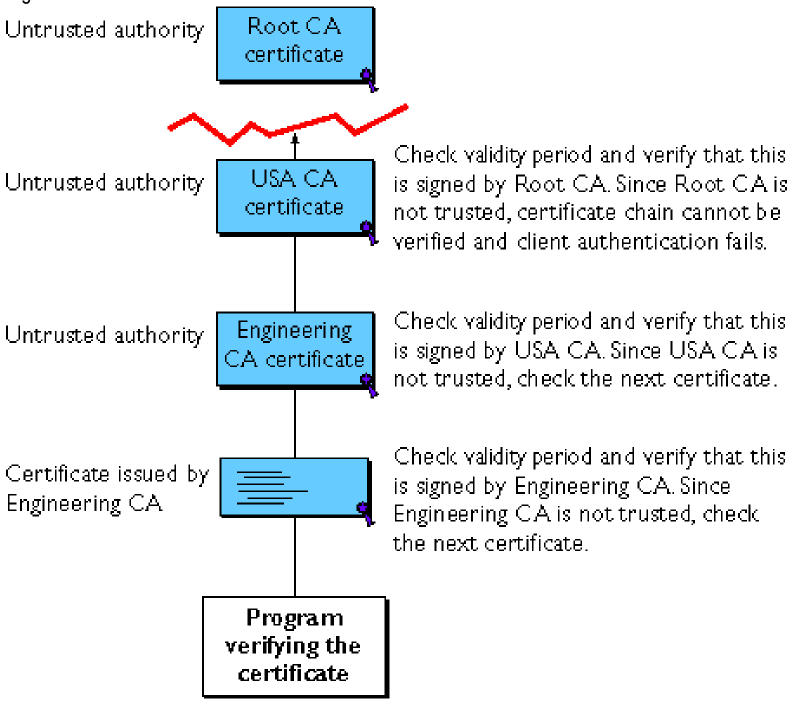
Figure 8 shows what happens when only Root CA is included in the verifier's local database. If a certificate for one of the intermediate CAs shown in Figure 8, such as Engineering CA, is found in the verifier's local database, verification stops with that certificate, as shown in Figure D.9.

Figure D.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 D.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 D.10A certificate chain that can't 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. For a more detailed description of the signature verification process in the context of SSL client and server authentication, see 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 networked environment is called the **public key infrastructure (PKI).** PKI management is complex topic beyond the scope of this document. The sections that follow introduce some of the specific certificate management issues addressed by Netscape 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 will be 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 will differ.

Similarly, different CAs have different procedures for issuing different kinds of certificates. In some cases the only requirement may be your email address. In other cases, your Unix or NT 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.

The Netscape Certificate Server, part of the Netscape family of products, 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. The Netscape Directory Server, part of the Netscape family of products, is fully integrated with the Netscape Certificate Server 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 will fail. 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 will fail 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 services.

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 will be 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.

Future versions of the Netscape Certificate Server will support the creation of customizable registration authorities.

Appendix

E

Introduction to SSL

This document introduces the Secure Sockets Layer (SSL) protocol. Originally developed by Netscape, SSL has been universally accepted on the World Wide Web for authenticated and encrypted communication between clients and servers.

- The SSL Protocol
- Ciphers Used with SSL
- The SSL Handshake

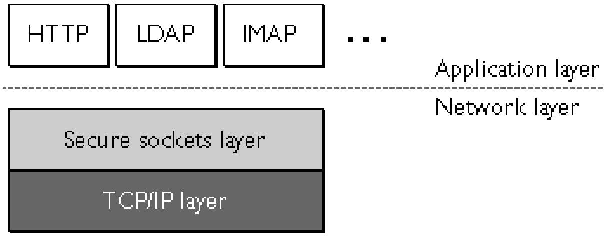
The new Internet Engineering Task Force (IETF) standard called Transport Layer Security (TLS) is based on SSL. This was recently published as an IETF Internet-Draft, The TLS Protocol Version 1.0. Netscape products will fully support TLS.

This document is primarily intended for administrators of Netscape server products, but the information it contains may also be useful for developers of applications that support SSL. The document assumes that you are familiar with the basic concepts of public-key cryptography, as summarized in the companion document 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 email servers.

Figure E.1 SSL runs above TCP/IP and below high-level application protocols



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.

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, company policies regarding acceptable encryption strength, and government restrictions on export of SSL-enabled software. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suites they will 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 will 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.

Some organizations may want to disable the weaker ciphers to prevent SSL connections with weaker encryption. However, due to U.S. government restrictions on products that support anything stronger than 40-bit encryption, disabling support for all 40-bit ciphers effectively restricts access to network browsers that are available only in the United States (unless the server involved has a special Global Server ID that permits the international client to “step up” to stronger encryption).

To serve the largest possible range of users, it’s a good idea for administrators to enable as broad a range of SSL cipher suites as possible. That way, when a domestic client or server is dealing with another domestic server or client, respectively, it will negotiate the use of the strongest ciphers available. And when an domestic client or server is dealing with an international server or client, it will negotiate the use of those ciphers that are permitted under U.S. export regulations.

However, since 40-bit ciphers can be broken relatively quickly, administrators whose user communities can use stronger ciphers without vilating export restrictions should disable the 40-bit ciphers if they are concerned about access to data by eavesdroppers.

Note **Netscape Console does not support all of the cipher suites supported by Netscape clients and servers. To ensure that Netscape Console can control an SSL-enabled server, the server must enable at least one of the following 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

Cipher Suites With RSA Key Exchange

Table E.1 lists the cipher suites supported by SSL that use the 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

Table E.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. Permitted for deployments within the United States only. This cipher suite is appropriate for banks and other institutions that handle highly sensitive data. Netscape Console does not support this cipher suite.	**Triple DES, which supports 168-bit encryption, with SHA-1 message authentication.** 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 FIPS-compliant. Both SSL 2.0 and SSL 3.0 support this cipher suite.

Table E.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. Permitted for deployments within the United States only. 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 (Data Encryption Standard), 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. Netscape Console supports only the SSL 3.0 version of this cipher suite.
	RC2 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 (Data Encryption Standard), with 168-bit encryption. RC4 and RC2 128-bit encryption permits approximately 3.4 * 10^{38} possible keys, making them very difficult to crack. RC2 ciphers are slower than RC4 ciphers. This cipher suite is supported by SSL 2.0 but not by SSL 3.0. Netscape Console does not support his cipher suite.
	DES, which supports 56-bit encryption, with 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. Netscape Console does not support this cipher suite.

Table E.1 Cipher suites supported by the SSL protocol that use the RSA key-exchange algorithm

Strength category and recommended use	Cipher suites
Exportable cipher suites. These cipher suites are not as strong as those listed above, but may be exported to most countries (note that France permits them for SSL but not for S/MIME). They provide the strongest encryption available for exportable products.[a]	**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. Netscape 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. Netscape Console supports only the SSL 3.0 version of this cipher suite.
Weakest cipher suite. This cipher suite provides authentication and tamper detection but no encryption. Server administrators must be careful about enabling it, however, because data sent using this cipher suite is not encrypted and may be accessed by eavesdroppers.	**No encryption, MD5 message authentication only.** 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.

[a] 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.

FORTEZZA Cipher Suites

Table E.2 lists additional cipher suites supported by Netscape products with FORTEZZA. for SSL 3.0. FORTEZZA is an encryption system used by U.S. government agencies to manage sensitive but unclassified information. It provides a hardware implementation of two classified ciphers developed by the federal government: FORTEZZA KEA and SKIPJACK. FORTEZZA ciphers for

SSL use the Key Exchange Algorithm (KEA) instead of the RSA key-exchange algorithm mentioned in the preceding section, and use FORTEZZA cards and DSA for client authentication.

Table E.2 FORTEZZA cipher suites supported by Netscape products with FORTEZZA for SSL 3.0

Strength category and recommended use	Cipher suites
Strong FORTEZZA cipher suites. Permitted for deployments within the United States only. These cipher suites support encryption that is strong enough for most business or government needs. Netscape Console does not support these cipher suites.	**RC4 with 128-bit encryption and SHA-1 message authentication.** Like RC4 with 128-bit encryption and MD5 message authentication, this cipher is one of the second strongest ciphers after Triple DES. It permits approximately 3.4 * 10^{38} possible keys, making it very difficult to crack. This cipher suite is supported by SSL 3.0 but not by SSL 2.0. **RC4 with SKIPJACK 80-bit encryption and SHA-1 message authentication.** The SKIPJACK cipher is a classified symmetric-key cryptographic algorithm implemented in FORTEZZA-compliant hardware. Some SKIPJACK implementations support key escrow using the Law Enforcement Access Field (LEAF). The most recent implementations do not. This cipher suite is supported by SSL 3.0 but not by SSL 2.0.
Weakest FORTEZZA cipher suite. This cipher suite provides authentication and tamper detection but no encryption. Server administrators must be careful about enabling it, however, because data sent using this cipher suite is not encrypted and may be accessed by eavesdroppers. These cipher suites do not support Netscape Console.	**No encryption, SHA-1 message authentication only.** This cipher uses SHA-1 message authentication to detect tampering. This cipher suite is supported by SSL 3.0 but not by SSL 2.0. Netscape Console does not suppor this cipher suite.

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 document. However, the steps involved can be summarized as follows (assuming the use of the cipher suites listed in Cipher Suites With RSA Key Exchange):

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 (see Server Authentication for details). 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 (see Client Authentication for details). 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 any 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 will be 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 will be 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, Netscape 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 will be 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.

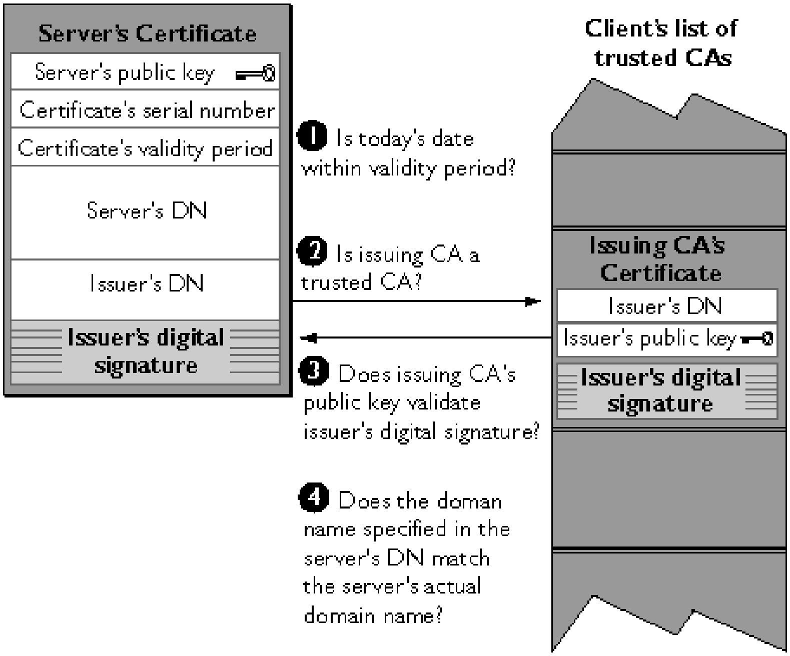
The sections that follow provide more details on Server Authentication and Client Authentication.

Server Authentication

Netscape's 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, 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 E.3. Although the fourth question is not technically part of the SSL protocol, it is the client's responsibility to support this requirement, which provides some assurance of the server's identity and thus helps protect against a form of security attack known as "man in the middle."

Figure E.2 How a Netscape server authenticates 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 E.2. This list determines which server certificates the client will accept. 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 will not be authenticated unless the client can verify a certificate chain ending in a CA that is on the list (see CA Hierarchies 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 will be 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.

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. Otherwise, the SSL session will be 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

As suggested in Step 4 above, the client application must check the server domain name specified in the server certificate against the actual domain name of the server with which the client is attempting to communicate. This step is necessry to protect against a manin-in-the-middle attack, which works as follows.

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 exhanged 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 sent 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 deteted. 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 thich a client is attempting to communicate--in addition to checking the validity of the certificate by performing the other stpes described in Server Authentication.

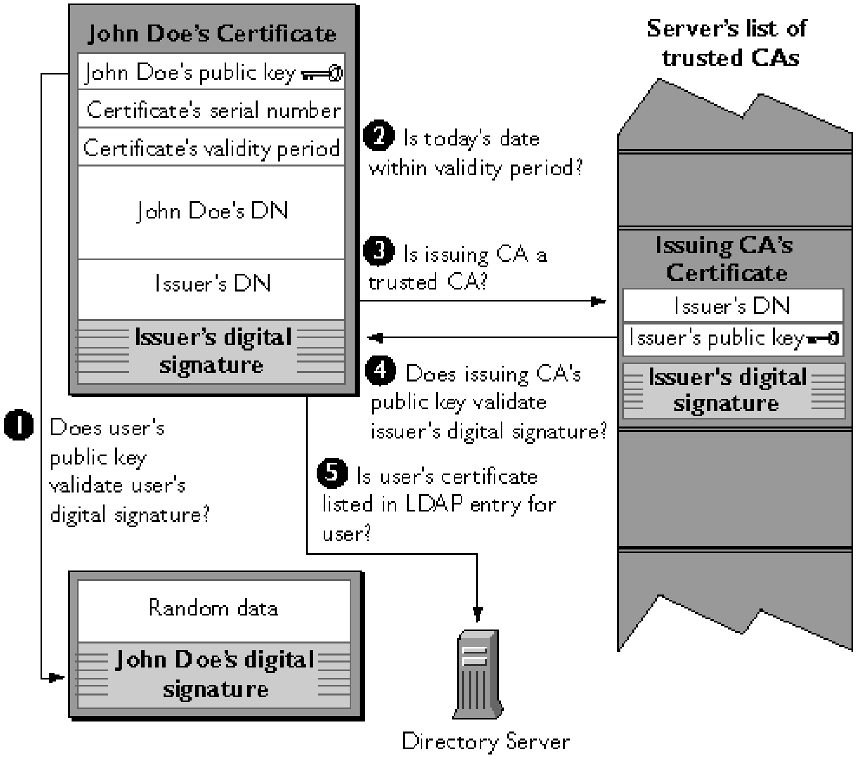
Client Authentication

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), 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 E.3. Although the fifth question is not part of the SSL protocol, Netscape 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 E.3 How a Netscape server authenticates a client certificate


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 E.3. This list determines which certificates the server will accept. 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 will not be authenticated unless the server can verify a certificate chain ending in a CA that is on the list (see CA Hierarchies 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. Netscape servers may optionally be configured to take 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 Netscape Certificate Server 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 will then refuse 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