

Managing Servers With iPlanet™ Console

Version 5.0

April 2001

Copyright © 2001 Sun Microsystems, Inc. Some preexisting portions Copyright © 2001 Netscape Communications Corporation. All rights reserved.

Sun, Sun Microsystems, the Sun logo, Java, JavaScript, iPlanet, and the iPlanet logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Netscape and the Netscape N logo are registered trademarks of Netscape Communications Corporation in the U.S. and other countries. Other Netscape logos, product names, and service names are also trademarks of Netscape Communications Corporation, which may be registered in other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Contains encryption software from RSA Data Security, Inc. Copyright © 1994 RSA Data Security, Inc. All rights reserved.

Contains the Taligent® International Classes™ from Taligent, Inc. and IBM Corp.

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

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

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

Copyright © 2001 Sun Microsystems, Inc. Pour certaines parties préexistantes, Copyright © 2001 Netscape Communication Corp. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, Java, JavaScript, iPlanet, et le logo iPlanet sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d'autre pays. Netscape et le logo Netscape N sont des marques déposées de Netscape Communications Corporation aux Etats-Unis et d'autre pays. Les autres logos, les noms de produit, et les noms de service de Netscape sont des marques déposées de Netscape Communications Corporation dans certains autres pays. UNIX est une marque déposée aux Etats-Unis et dans d'autre pays et licenciée exclusivement par X/Open Company, Ltd.

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

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

Contents

About This Guide	13
What's in This Guide	13
Conventions Used in This Guide	13
Viewing This Guide Online	15
To View This Manual From iPlanet Console or Administration Server	15
To View This Manual From Another Product	15
Getting Additional Help	15
To Get Context-Sensitive Help	16
To Search this Guide's Index	16
To Open the Product Homepage	17
Part 1 Overview of iPlanet Console	19
Chapter 1 iPlanet Console and Administration Server	21
Chapter 2 Installing iPlanet Servers and Console	25
The Setup Program	25
Installing a New Server	26
Directory Server Must Be Installed First	26
Administration Server Is Required in Each Server Root	26
Installation Modes	26
Express	27
Typical	27
Custom	27
Installing iPlanet Console as a Stand-Alone Application	27
To Install iPlanet Console as a Stand-Alone Application on UNIX System	27
To Install iPlanet Console as a Stand-Alone Application on Windows NT Systems	28
Upgrading to Version 5.0	30
Upgrading Administration Server and Console	30

To Upgrade on UNIX Systems	30
To Upgrade on Windows NT Systems	32
Upgrading a Stand-Alone Version of iPlanet Console	33
To Upgrade a Stand-Alone Version of iPlanet Console on UNIX Systems	34
To Upgrade a Stand-Alone Version of iPlanet Console on Windows NT Systems	34
Silent Installation	36
Performing a Silent Installation	36
To Save Your Installation Answers	36
To Perform a Silent Installation	37
Uninstallation	37
Uninstalling an iPlanet Server	37
To Uninstall an iPlanet Server on UNIX Systems	37
To Uninstall an iPlanet Server on Windows NT Systems	38
Silent Uninstallation	39
To Perform a Silent Uninstallation on UNIX Systems	39
To Perform a Silent Uninstallation on Windows NT Systems	40

Part 2 iPlanet Console Basics 41

Chapter 3 Using iPlanet Console	43
Starting iPlanet Console and Logging In	43
Starting iPlanet Console	43
To Start iPlanet Console on UNIX Systems	43
To Start iPlanet Console on Windows NT Systems	44
Logging In to iPlanet Console With a User Name and Password	45
To Log in to iPlanet Console With a User Name and Password	45
Logging In to iPlanet Console Using Client Authentication	46
To Request and Install a New Client Certificate	47
To Make Your Client Certificate Available to iPlanet Console on UNIX Systems	47
To Make Your Client Certificate Available to iPlanet Console on Windows NT	47
To Establish a Secure Connection With an Instance of Administration Server	48
A Tour of iPlanet Console	49
iPlanet Console Menus	49
iPlanet Console Tabs	50
The Servers and Applications Tab	50
The Administration Domain	51
To Create an Administration Domain	52
To Modify an Administration Domain	53
To Remove an Administration Domain	54
Customizing iPlanet Console	54
Storing Display Settings	54

To Change Where Display Settings Are Stored	55
To Reset Display Settings to Their Default Values	55
Setting Display Fonts	55
To Create a Font Profile	55
To Edit an Existing Font Profile	56
To Rename a Font Profile	57
To Use a Font Profile	57
To Remove a Font Profile	57
Customizing the Main Window	58
To Customize the Main Window	58
Customizing Tables	59
To Change Column Position in a Table	59
To Change the Width of Columns in a Table	60
Creating Custom Views of the Navigation Tree	61
To Create a Custom View of the Navigation Tree	61
Working With Custom Views	63
To Switch to a Custom View	64
To Edit a Custom View	64
To Rename a Custom View	64
To Set Access Permissions for a Public View	65
To Delete a Custom View	65
Administration Express	65
Accessing Administration Express	66
To Open Administration Express	66
Using Administration Express	67
To Start or Stop a Server Instance From Administration Express	67
To View Basic Server Information From Administration Express	68
To View Access and Error Logs From Administration Express	68
Setting the Refresh Rate for Administration Express	68
To Set the Refresh Rate for Administration Express	68
Chapter 4 Servers in iPlanet Console	71
Working With Earlier Netscape Servers	71
Adding a Pre-4.0 Server to the Tree	71
To Add a Pre-4.0 Server to the Navigation Tree	72
Migrating from a Pre-4.0 Server to a Newer Server	73
To Migrate From a Pre-4.0 Server to a Newer Version	74
Working With iPlanet Servers	75
Opening a Server Management Window	75
To Open an iPlanet Server Management Window	75
Creating a New Server Instance	76
To Create a New Server Instance	77
Modifying Host, Server Group, and Instance Information	77

To Modify Host, Server Group, and Instance Information	77
Cloning a Server	78
To Clone Server Settings to Another Server	78
Removing a Server Instance	78
To Remove a Server Instance	78
Uninstalling an iPlanet Server	79
Merging Configuration Data From Two Directory Servers	79
To Merge Configuration Data From Two Directory Servers	82
Chapter 5 User and Group Administration	83
Interacting with Directory Server	83
Using Distinguished Names	83
Distinguished Names, Attributes, and Syntax	84
Distinguished Names	84
Attributes	86
DN and Attribute Guidelines and Syntax	87
Locating a User or Group in the Directory	89
To Locate Users or Groups in the Directory	89
Choosing a Different Directory to Search	90
To Change the Directory to Search	90
Creating New Directory Entries	91
Users	91
To Create a New User Entry in the Directory	91
The User's Preferred Language	94
Administrators	95
To Create an Administrator	95
Specifying Windows NT and UNIX Systems Options	96
To Enable Windows NT and UNIX Systems Panels for an Individual User	96
To Enable Windows NT and UNIX Systems Panels for All New Users	96
To Set Windows NT and UNIX Systems Options and Attributes for a New User	97
Groups	99
To Create a Static Group in the Directory	100
To Add Users to the Configuration Administrators Group	101
To Create a Dynamic Group	102
To Create a Certificate Group	104
Organizational Units	106
To Create a New Organizational Unit	106
Modifying Existing Directory Entries	107
Updating User and Group Entries	107
To Edit a User or Group Entry in the Directory	107
To Change a User Password	107
To Change the Configuration Administrator's User Name or Password	107
To Change the Administration Server Administrator's User Name or Password	109

To Remove a User, Group, or Organizational Unit From the Directory	109
--	-----

Part 3 Using iPlanet Administration Server 111

Chapter 6 Administration Server Basics	113
Restarting Administration Server	113
To Restart the Server From iPlanet Console	113
To Restart the Server From the Command Line	114
UNIX Systems	114
Windows NT Systems	114
To Restart the Server From the NT Control Panel	115
Stopping Administration Server	115
To Stop the Server From iPlanet Console	115
To Stop the Server From the Command Line	115
UNIX Systems	115
Windows NT Systems	115
To Stop the Server From the NT Control Panel	116
Logging Options	116
To View the Access Log	117
To View the Error Log	117
To Change Where Logs Are Stored	118
The iPlanet Administration Page	118
To Access the Administration Page	119
Chapter 7 Administration Server Configuration	121
Network Settings	121
To Configure Network Settings	122
Access Settings	123
To Set Administration Server Access Settings	124
Encryption Settings	125
To Request and Install a Certificate for Administration Server	125
To Activate SSL on Administration Server	125
Directory Settings	127
The Configuration Directory	127
Changing the Host or Port Number	127
To Change the Host or Port Number	128
The User Directory	129
User Directory Settings	129
User Authentication and Directory Failover Support	130
Changing User Directory Settings for a Domain	130
To Change the User Directory Settings for a Domain	130

To Change User Directory Settings for a Server Group	132
--	-----

Chapter 8 Administration Server Command-Line Tools	135
admconfig	135
Syntax	135
Options	136
Tasks and Their Arguments	137
Examples	144
admin_ip.pl	145
Usage	145
ldapsearch, ldapmodify, and ldapdelete	146
sec-activate	146
Syntax	146
Example	146
sec-migrate	146
Syntax	147
modutil	147
Syntax	147
Tasks and Options	148
Usage	153
JAR Information File	155
JAR Information File Syntax	157
Examples of Using modutil	162

Part 4 Advanced Server Management 169

Chapter 9 Access Control	171
Overview of Access Control	171
Examples of Access Control	172
Setting Access Permissions For Servers	173
To Set Access Permissions for a Server in the Navigation Tree	173
Working With Access Control Instructions	174
What's in an ACI	175
Target	175
Permissions	175
Bind Rules	175
Using the ACI Manager and ACI Editor	176
To Specify What You Want an ACI to Apply To	176
To Create a New ACI With the Visual ACI Editor	177
To Create a New ACI With the Manual ACI Editor	181
To Edit an Existing ACI With the ACI Editor	182

To Remove an ACI	182
Chapter 10 Using SSL and TLS with iPlanet Servers	183
The SSL and TLS Protocols	183
SSL and TLS Ciphers	184
Choosing SSL and TLS Ciphers	184
Preparing to Use SSL and TLS Encryption	185
Using External Security Devices	185
Slots and Security Devices	185
To Install an External Security Device	186
To Remove an External PKCS #11 Module	186
Obtaining and Installing a Server Certificate	187
SSL Certificates	187
Preparing to Set Up SSL and TLS	188
Setting up SSL or TLS With an Internal Security Device	188
Setting up SSL or TLS With an External Security Device	188
Setting Up SSL With Internal and External Security Devices	188
Generating a Server Certificate Request	188
To Generate a Certificate Request	188
Sending a Server Certificate Request	190
To Send a Server Certificate Request as Email	190
Installing the Certificate	191
To Back Up a Certificate	191
To Install a Server Certificate	191
To Install a CA Certificate or Server Certificate Chain	192
Backing Up and Restoring Your Certificate Database	193
To Back Up Your Certificate Database	193
To Restore Your Certificate Database From a Backup	193
Activating SSL	194
To Activate SSL on an iPlanet Server or a Netscape 4.x Server	194
Managing Server Certificates	196
Renewing a Certificate	196
To Check a Certificate Expiration Date	196
To Generate a Certificate Renewal Request	196
Changing the CA Trust Options	198
To Change the CA Trust Options	198
Changing Security Device Passwords	200
To Change a Security Device Password	200
Managing Certificate Lists	200
To Obtain a CRL or CKL From a CA	201
To View, Add, or Delete a CRL or CKL	201
Using Client Authentication	202
How Client Authentication Works	202

Preparing to Use Client Authentication	203
The certmap.conf File	204
DNComps	205
FilterComps	205
VerifyCert	206
CmapLdapAttr	206
Library	206
InitFn	206
Custom Properties	207
Editing the certmap.conf File	207
To Edit the certmap.conf File	207
Example certmap.conf Mappings	208
Example of a Default Mapping	208
Example of an Additional Mapping	208
Example of a Mapping With an Attribute Search	209
Using Client Authentication Between Servers	210
To Set Up Client Authentication Between Servers	210
Client Authentication for Users	211
To Set Up Client Authentication for Users	211
Chapter 11 Using SNMP to Monitor Servers	213
SNMP Basics	213
How SNMP Works	215
iPlanet MIBs	216
The Administration Server MIB	216
Types of SNMP Messages	217
Network Management Station-Initiated Communication	218
Server-Initiated Communication	218
Setting Up SNMP on UNIX Systems	218
Using a Proxy SNMP Agent on UNIX Systems	220
Installing and Starting the Proxy SNMP Agent	220
To Install the SNMP Proxy Agent	221
To Start the SNMP Proxy Agent	221
To Restart the Native Agent	221
Reconfiguring a Native Agent on UNIX Systems	222
Configuring the Master Agent on UNIX Systems	222
Community Strings	222
Trap Destinations	223
Configuring the Master Agent using iPlanet Console	223
To Add, Edit, or Remove a Community String using iPlanet Console	223
To Add, Edit, or Remove a Trap Destination	225
Manually Configuring the Master Agent	226
To Configure the Master SNMP Agent Manually	226

Editing the Master Agent Config File	227
Defining sysContact and sysLocation Variables	227
Starting the Master Agent on UNIX Systems	228
Starting the Agent Using iPlanet Console	228
To Start the Master Agent Using iPlanet Console	228
Starting the Agent From the Command Line	229
To Start the Agent on the Standard Port	229
To Start the Agent on a Non-Standard Port Using the Config File	229
To Start the Agent on a Non-Standard Port Using System Services	230
Enabling the Subagent on UNIX Systems	230
Using the Windows NT SNMP Service	230
To Set Up SNMP on Windows NT Systems	230

Part 5 Appendixes 231

Appendix A Fortezza	233
How It Works	233
How Fortezza Crypto Cards Are Certified	234
Fortezza Keys, Certificates, and Encryption	235
CRLs and CKLs	235
Encryption Algorithms	235
SKIPJACK	235
SSL Protocol	235
RC4 Encryption	235
NULL Encryption	235
Enabling Fortezza	236
To Enable Fortezza on Administration Server	236
Appendix B Introduction to Public-Key Cryptography	237
Internet Security Issues	237
Encryption and Decryption	239
Symmetric-Key Encryption	239
Public-Key Encryption	240
Key Length and Encryption Strength	242
Digital Signatures	242
Certificates and Authentication	244
A Certificate Identifies Someone or Something	244
Authentication Confirms an Identity	245
Password-Based Authentication	246
Certificate-Based Authentication	248
How Certificates Are Used	250

Types of Certificates	250
SSL Protocol	252
Signed and Encrypted Email	252
Form Signing	253
Single Sign-On	253
Object Signing	254
Contents of a Certificate	255
Distinguished Names	255
A Typical Certificate	256
How CA Certificates Are Used to Establish Trust	258
CA Hierarchies	259
Certificate Chains	260
Verifying a Certificate Chain	261
Managing Certificates	264
Issuing Certificates	265
Certificates and the LDAP Directory	266
Key Management	266
Renewing and Revoking Certificates	267
Registration Authorities	268
Appendix C Introduction to SSL	269
The SSL Protocol	269
Ciphers Used With SSL	271
Cipher Suites With RSA Key Exchange	272
Fortezza Cipher Suites	274
The SSL Handshake	275
Server Authentication	278
Man-in-the-Middle Attack	280
Client Authentication	280
Glossary	285
Index	297

About This Guide

Managing Servers With iPlanet™ Console provides background information that system architects and administrators need to successfully install and manage iPlanet servers in their enterprise. Read about iPlanet server basics here before you begin installing and configuring servers in your enterprise.

What's in This Guide

This book provides information you need to use iPlanet servers. It is divided into the following parts:

- Part 1, “Overview of iPlanet Console”
- Part 2, “iPlanet Console Basics”
- Part 3, “Using iPlanet Administration Server”
- Part 4, “Advanced Server Management”
- Part 5, “Appendixes”

Conventions Used in This Guide

The following typographical conventions are used in this guide:

`Monospaced font`

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

Boldface

In UI reference material, boldface type identifies window elements such as input areas and checkboxes.

Italic

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

TIP Tips are useful information that can help you save time.

NOTE Notes mark important information. Make sure you read the information before continuing with a task.

CAUTION Cautions alert you to potentially problematic situations, and tell you how to avoid them.

[]

Square brackets enclose commands that are optional. You can choose to omit any text that appears in square brackets.

/

Forward slashes are used to separate directories in a path. If you use the Windows NT operating system, you may be more familiar with paths containing back slashes (\). NT supports both types of slashes; you can use whichever you prefer.

>

Forward angle brackets are used to indicate menu hierarchies. For example, the text “from the Console menu, choose Security > Manage Certificates” means that you should open the Console menu, select the Security item to open its submenu, and then choose the Manage Certificates item from that submenu.

“Start”

In Windows NT -related sections of this guide, “Start” typically refers to the Windows NT Start menu button. For example, “click Start, and then choose Programs > iPlanet Server Products > iPlanet Console 5.0” means that you should click the Windows NT Start menu button, and then select Programs > iPlanet Server Products > iPlanet Console 5.0.

UNIX

Marks text that applies only to UNIX users.

NT

Marks text that applies only to Windows NT users.

Viewing This Guide Online

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

To View This Manual From iPlanet Console or Administration Server

1. From the Help menu, choose Contents or press the F1 key.

A browser window opens and displays an HTML version of the table of contents for this manual. Click a link to go to a chapter or section.

To View This Manual From Another Product

1. From the server management window's Help menu, choose Documentation Resources.

A browser window opens and displays a Documentation Resources page.

2. Click *Managing Servers With iPlanet Console* to view an HTML version of this manual's table of contents. Click a link to go to a chapter or section.

Getting Additional Help

The following types of help are available from within iPlanet Console:

- Context-sensitive help
- A searchable version of this guide's index
- A Documentation Resources page with product-related links.

This section shows you how to access these resources.

To Get Context-Sensitive Help

1. Click a Help button.

You will see a browser window with information about the screen you are viewing.

2. If you need further assistance, click one of the following links at the top or bottom of the help screen:

Help Topics and Procedures. This displays a list of all available help topics and procedures for the product you're working in.

Manual Contents. This displays the table of contents of the manual for the product you're working in.

Manual Index. This displays the index of the manual for the product you're working in.

Documentation Resources. This displays the Documentation Resources page, which contains links to documentation for the product you're using.

To Search this Guide's Index

1. From the Help menu, choose Index.

This opens the Search Index dialog box, an interface used for searching this guide's index. The text field at the top of the dialog box accepts a search term, the middle frame shows an alphabetical list of all indexed terms, and the bottom frame is used to show topics.

2. Enter a search term in the top field of the search interface.

If the index contains your search term, you will see it highlighted in the alphabetical list. If your search term is not found, the closest match is highlighted.

3. Click the desired topic from the bottom frame.

These topics are links to sections of this guide. Clicking one opens a browser displaying the appropriate section.

4. To dismiss the Search Index dialog box, click Close.

To Open the Product Homepage

- From the Help menu, choose Documentation Home.

A browser window opens containing a list of iPlanet Console-related links. Alternatively, you can access this page by clicking Documentation Resources from within context-sensitive help.

Getting Additional Help

Overview of iPlanet Console

Chapter 1, “iPlanet Console and Administration Server”

Chapter 2, “Installing iPlanet Servers and Console”

iPlanet Console and Administration Server

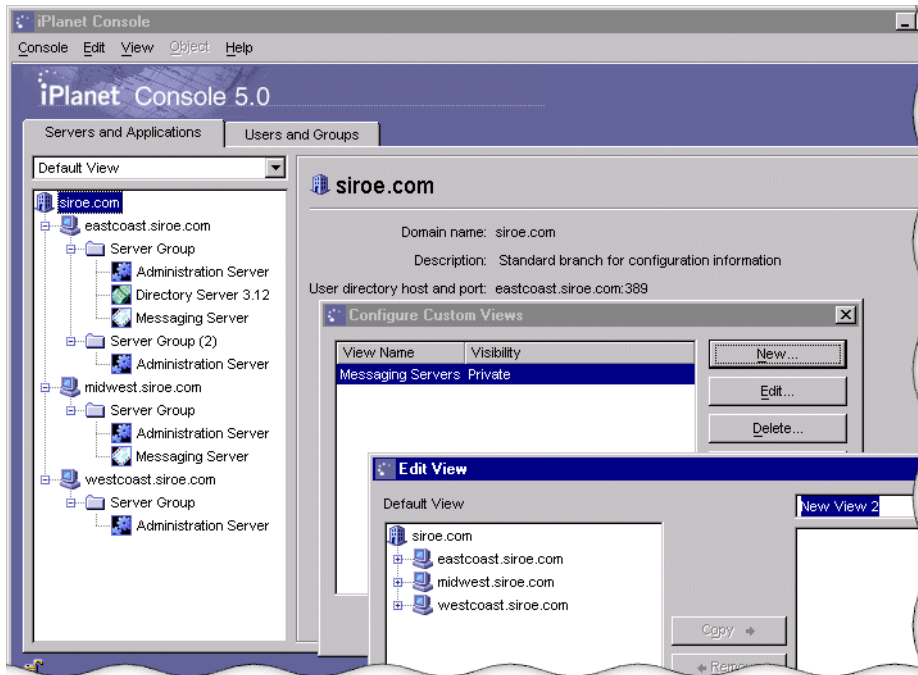
iPlanet Console 5.0 and Administration Server 5.0 are two parts of a system that lets you manage iPlanet software and users in your enterprise. This chapter presents a high-level overview of what this system is and how you can use it to work with resources across your network.

In order to run most iPlanet software, you must first install iPlanet Directory Server. By default, when you do this, iPlanet Console and Administration Server are automatically installed for you. Although iPlanet Directory Server, iPlanet Console, and iPlanet Administration Server work tightly with one another, each plays a specific role in the management of servers, applications, and users.

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

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

Figure 1-1 The iPlanet Console Interface



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

When you install an iPlanet product in a new folder, iPlanet Administration Server is installed for you. If you install additional products in the same folder, they can use the instance of iPlanet Administration Server that is already there. If a product includes a newer version of iPlanet Administration Server and iPlanet Console than the versions in the root folder, the installer updates the folder with the latest versions. iPlanet Administration Server and iPlanet Console are backward compatible; all existing Netscape or iPlanet servers will continue to work normally.

The system for managing iPlanet products works as follows:

iPlanet Console lets you manage resources (servers or applications) as well as add or edit user information. When you use iPlanet Console to manage resources, Console sends HTTP requests to the instance of iPlanet Administration Server that controls the resource. Upon receiving these requests, the instance of iPlanet

Administration Server executes programs that perform the requested tasks. For example, iPlanet Administration Server can execute programs to modify the server and application settings that are stored in the configuration directory or to change the port number that a server listens to.

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

Figure 1-2 A Simple System With iPlanet Console

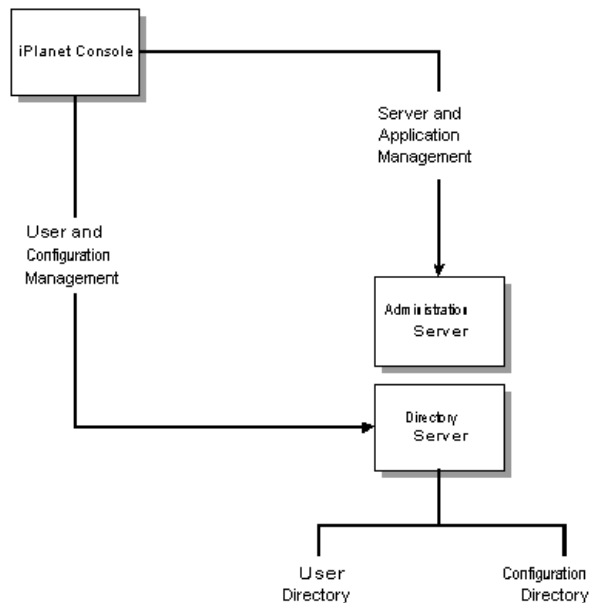
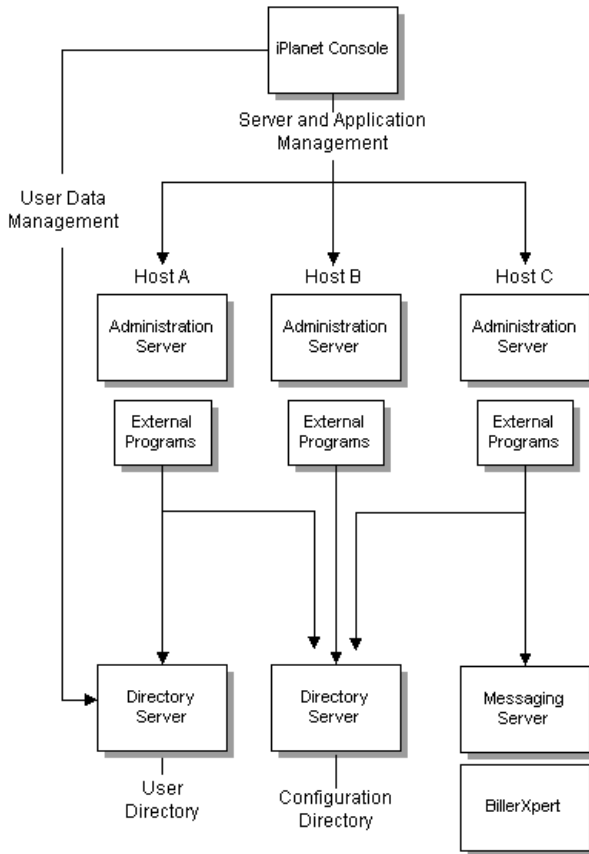


Figure 1-2 shows an example of a relatively simple system. As your enterprise grows and your needs change, you have the flexibility to add additional hosts and servers. Even when you install new hardware and software, you can continue to use a single instance of iPlanet Console to manage your network.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

Figure 1-3 A More Complex System With iPlanet Console



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

If you would like to learn more about how iPlanet Console works before installing the product, see “A Tour of iPlanet Console” on page 49.

Installing iPlanet Servers and Console

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

This chapter contains the following sections:

- The Setup Program
- Upgrading to Version 5.0
- Silent Installation
- Uninstallation

Each iPlanet server product has its own detailed installation instructions. To read these, see your server product documentation at <http://docs.iplanet.com>.

The Setup Program

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

- Install a new server or server component
- Install iPlanet Console as a stand-alone application
- Update a server

Installing a New Server

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

NOTE Each iPlanet and Netscape™ server has its own detailed installation instructions. Look for your server's documentation at <http://docs.iplanet.com>

Directory Server Must Be Installed First

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

For detailed information on installing Directory Server, see the server's documentation at <http://docs.iplanet.com>.

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

Administration Server Is Required in Each Server Root

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

NOTE Installing or upgrading iPlanet Console on Windows NT requires rebooting the machine at the end of the install process. The option to reboot is offered at the end of the setup program. If you choose not to reboot at the end of the install process you must remember to reboot later, before you use iPlanet Console.

Installation Modes

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

Express

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

Typical

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

Custom

Use this mode only if you'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. When using custom mode, you can specify every typical option as well as advanced ones such as the IP address of a host system.

Installing iPlanet Console as a Stand-Alone Application

You can install iPlanet Console as a stand-alone application on a machine local to you. Having iPlanet Console on your local machine allows you to manage servers on remote machines.

To Install iPlanet Console as a Stand-Alone Application on UNIX System

1. Download the compressed product binaries for iPlanet Console.

These are available at <http://www.ipplanet.com/downloads/patches/>

2. Extract the binaries into a new directory.
3. Run the Setup program by typing `setup`.

The first installation screen appears.

4. Proceed through the installation process. Here are the prompts you encounter with instructions about what to do:

Would you like to continue with installation? Enter `Yes`.

Do you agree to the license terms? Enter `Yes`.

Select the component you want to install. Enter `2` for iPlanet Console

Installation location. Enter the path to the folder where you want to install iPlanet Console. If the specified folder does not exist, the Setup program will create it for you.

5. Press Enter.

The Setup program installs iPlanet Console in the folder you specified.

Once installation is complete, you can run iPlanet Console by navigating to the folder you specified as the installation location, and then typing `startconsole`.

To Install iPlanet Console as a Stand-Alone Application on Windows NT Systems

1. Download the compressed product binaries for iPlanet Console.

The binary files are available at this location:

<http://www.iplanet.com/downloads/patches>

2. Extract the binaries into a new folder and run the `setup.exe` program.

The installation startup screen appears.



3. Proceed through the installation process. Here are the prompts you encounter with instructions about what to do:
4. **Do you accept all of the terms of the preceding license agreement?** Click Yes.
5. **Choose the type of Setup you prefer.** Select iPlanet Console.
6. **Installation directory.** Enter the location where you want to install iPlanet Console. If this folder does not exist, the Setup program asks if you want to create it.
7. Review your selections. If you need to make any changes, click Back and modify your choices.
8. Click Install.

The Setup program installs iPlanet Console in the specified folder.

9. When the installation is complete, click Finish.

Once installation is complete, you can run iPlanet Console by clicking Start, and then choosing Programs > iPlanet Server Products > iPlanet Console 5.0.

Upgrading to Version 5.0

If you already have versions of Netscape Console and Administration Server installed on your system, you can upgrade to iPlanet Console 5.0. This section contains instructions for performing the following upgrades:

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

NOTE The instructions presented in this section apply only when upgrading iPlanet or Netscape Administration Server and Console. If you want to upgrade a different iPlanet or Netscape product, please refer to the installation instructions for the upgraded version of that product. You can find most installation instructions at <http://docs.iplanet.com>.

Upgrading Administration Server and Console

To upgrade Netscape Administration Server and Console to iPlanet Administration Server and Console 5.0, follow the directions for your operating system.

To Upgrade on UNIX Systems

1. Download the compressed product binaries for iPlanet Administration Server and Console.

The binary files are available at this location:

<http://www.iplanet.com/downloads>.

2. Extract the binaries files into a new folder.
3. Run the Setup program by typing `setup`.

The first installation screen appears.

4. Proceed through the installation process. Here are the prompts you encounter with instructions about what to do:

Would you like to continue with installation? Press Enter for Yes.

Do you agree to the license terms? Enter `Yes`.

Select the component you want to install Enter `1` for iPlanet Servers.

Choose an installation type Enter `2` for Typical.

Installation location Enter the location where Administration Server is currently installed.

If Administration Server was installed with another Netscape or iPlanet server, enter the path to that product's server root. For example, if you installed Netscape Directory Server 4.1 in the `/usr/netscape/server4` folder, then you would enter `/usr/netscape/server4` as your installation location.

Specify the components you wish to install Press Enter (for All)

(Core Components) Specify the components you wish to install Choose all three core components by entering `1, 2, 3`.

(Administration Services) Specify the components you wish to install Choose both components by entering `1, 2`.

Computer name Enter the fully qualified hostname of your computer. For example, `eastcoast.siroe.com`.

System User Enter the user ID that iPlanet Administration Server is currently running as. The server will continue to run as this user.

System Group Enter the UNIX group to which the System User belongs.

Configuration Admin ID or DN Enter the user ID or distinguished name of the administrator who is currently authorized to access the configuration directory.

Password Enter the password for the user specified by the Configuration Admin ID or DN.

5. Press Enter.

The installer replaces your existing Administration Server and Console with the new versions of the software.

Once installation is complete, you can run iPlanet Console by navigating to the folder you specified as the Install location, and then typing `startconsole`.

To Upgrade on Windows NT Systems

1. Download the compressed product binaries for iPlanet Administration Server and Console.

The binary files are available at this location:

<http://www.iplanet.com/downloads>

2. Extract the binaries into a new folder and run the `setup.exe` program.

The installation startup screen appears.



3. Click Next.

4. Proceed through the installation process. Here are the prompts you encounter with instructions about what to do:

Do you accept all of the terms of the preceding license agreement? Click Yes.

Choose the type of Setup you prefer Select iPlanet Servers.

(Type of Installation) Choose the type of Setup you prefer Select Typical.

Installation directory Enter the location where iPlanet Administration Server is currently installed.

If Administration Server was installed with another Netscape or iPlanet server, enter the path to that product's server root. For example, if you installed Netscape Directory Server 4.1 in the C:\Netscape\Server4 folder, you would enter C:\Netscape\Server4 as your installation location.

Select the products you want to install Both boxes are checked, by default.

User ID or Distinguished Name Enter the user ID or distinguished name of the administrator who is currently authorized to access the configuration directory.

Password Enter the password for the user ID or distinguished name entered above.

5. Review your selections. If you need to make any changes, click Back and modify your choices.
6. Click Next.

The Setup program replaces your existing Administration Server and Console with version 5.0.

7. When the installer completes, click Finish.

Once installation is complete, you can run iPlanet Console by clicking Start, and then choosing Programs > iPlanet Server Products > iPlanet Console 5.0.

Upgrading a Stand-Alone Version of iPlanet Console

If you have installed a stand-alone version of iPlanet Console, you can upgrade it to version 5.0.

To Upgrade a Stand-Alone Version of iPlanet Console on UNIX Systems

1. Download the compressed product binaries for iPlanet Console.

The binary files are available at this location:

<http://www.iplanet.com/downloads>

2. Extract the binaries into a new folder.
3. Run the Setup program by typing `setup`.

The first installation screen appears.

4. Proceed through the installation process. Here are the prompts you encounter, with instructions about what to do:

Would you like to continue with installation? Press Enter for Yes.

Do you agree to the license terms? Enter `Yes`.

Select the component you want to install Enter `2` for iPlanet Console.

Installation location Enter the location where iPlanet Console is currently installed.

5. Press Enter.

The installer replaces your existing version of iPlanet Console with the new version of the software.

Once installation is complete, you can run iPlanet Console by navigating to the folder you specified as the installation location, and then typing `startconsole`.

To Upgrade a Stand-Alone Version of iPlanet Console on Windows NT Systems

1. Download the compressed product binaries for iPlanet Console.

The binary files are available at this location:

<http://www.iplanet.com/downloads>

2. Extract the binaries into a new folder and run the `setup.exe` program.

The installation startup screen appears.



3. Click Next.
4. Proceed through the installation process. Here are the prompts you encounter with instructions about what to do:

Do you accept all of the terms of the preceding license agreement? Click Yes.

Choose the type of Setup you prefer. Select iPlanet Console.

Installation directory. The installer will automatically supply the location where Console is currently installed.

5. Review your selections. If you need to make any changes, click Back and modify your choices.
6. Click Install.

The Setup program replaces your existing version of iPlanet Console with the new version of the software.

7. When the installer completes, click Finish.

Once installation is complete, you can run iPlanet Console by clicking Start, and then choosing Programs > iPlanet Server Products > iPlanet Console 5.0.

Silent Installation

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

Performing a Silent Installation

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

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

To Save Your Installation Answers

1. From the system prompt, run the Setup program by typing `setup -k`.

The `-k` flag instructs the Setup program to store your answers to installation questions.

2. Perform your installation or upgrade.

The answers that you specify in response to installation and upgrade questions are stored in the `setup/install.inf` file which is contained in the destination directory that you specify during installation.

3. If you plan to perform multiple silent installations using different sets of installation answers, rename `install.inf` to a name that clearly identifies the set of installation specification you have chosen and then repeat this procedure.
4. Repeat steps 1 through 3 for each set of installation specifications you need.

For more details on installation, see "The Setup Program," which begins on page 25.

To Perform a Silent Installation

1. Make any necessary changes to the file or files containing your installation answers.
2. Copy the installation answer file or files to the directory containing the Setup program.
3. From the system prompt, run the Setup program by typing `setup -s -f filename`.

The `-s` flag instructs the Setup program to perform a silent installation. The `-f` flag tells the Setup program to use the answer file specified by `filename`.

On UNIX, Silent Installation outputs some status messages and alerts. Complete status information is written to the `setup/setup.log` file which is contained in the destination directory that you indicate during installation.

On Windows NT, Silent Installation does not produce any status messages or alerts. All status information is written to the `setup/setup.log` file which is contained in the destination directory that you indicate during installation.

For detailed information on how a particular server uses Silent Installation, see that server's documentation.

Uninstallation

If you are no longer using an iPlanet server, you can uninstall it. Uninstallation completely removes a server from your computer. The server will not be accessible and you will lose all settings.

Uninstalling an iPlanet Server

The following procedures show you how to uninstall an iPlanet server on UNIX and Windows NT.

To Uninstall an iPlanet Server on UNIX Systems

1. In the server root, type `uninstall`.

The first uninstallation screen appears.

2. Proceed through the uninstallation process. Here are the prompts you encounter with instructions about what to do. Depending on the selections you make, you may see additional prompts:

Select the components you wish to uninstall Select the components to uninstall or press Enter (for All) to remove all listed software.

Configuration Admin ID or DN Enter the user ID or distinguished name of the administrator who is currently authorized to access the configuration directory.

Password Enter the password for the user specified by the Configuration Admin ID or DN.

3. Press Enter.

The uninstall program removes the selected software. If the uninstall program cannot remove all files in the server root, it prints a message to the screen. To remove any remaining files, go to the server root and delete the files manually.

To Uninstall an iPlanet Server on Windows NT Systems

1. Click Start, and then choose Settings > Control Panel.
2. Double-click Add/Remove Programs.
Alternatively, you can run `uninst.exe` from the server root.
3. In the Add/Remove Program Properties window, click the Install/Uninstall tab.
4. Select iPlanet Server Products 5.0, then click Remove.
5. In the iPlanet Uninstall window, select the iPlanet servers and components you want to uninstall.
6. If you want to specify which subcomponents of your iPlanet software to remove, highlight the installed product or component name and then click the Subcomponents button.

The Select Sub-components dialog appears. Select the subcomponents that you want to remove, then click Continue.

Select the components you wish to uninstall Select the components to uninstall or press Enter (for All) to remove all listed software.

Configuration Admin ID or DN Enter the user ID or distinguished name of the administrator who is currently authorized to access the configuration directory.

7. **Password** Enter the password for the user specified by the Configuration Admin ID or DN.
8. Click Uninstall.

The uninstall program removes the selected software. If the uninstall program cannot remove all files in the server root, it prints a message to the screen. To remove any remaining files, go to the server root and delete the files manually.

Silent Uninstallation

The Silent Uninstallation feature allows you to uninstall a product without providing answers to uninstallation questions.

To Perform a Silent Uninstallation on UNIX Systems

- From the system prompt, run the uninstallation program in silent mode by typing `uninstall -s`.

If the uninstallation program cannot contact the instance of Directory Server containing the configuration information for the product you are trying to uninstall, uninstallation will fail. In this case, no product files or configuration information will be removed. If you want the uninstallation program to remove the local product files regardless of whether it can contact the instance of Directory Server containing configuration information, run the uninstallation program by typing `uninstall -s -force`.

While it removes files, the uninstallation program outputs some status messages and alerts. When uninstallation is finished, you are returned to the system prompt.

To Perform a Silent Uninstallation on Windows NT Systems

- From the system prompt, run the uninstallation program in silent mode by typing `uninst -s`.

If the uninstallation program cannot contact the instance of Directory Server containing the configuration information for the product you are trying to uninstall, uninstallation will fail. In this case, no product files or configuration information will be removed. If you want the uninstallation program to remove the local product files regardless of whether it can contact the instance of Directory Server containing configuration information, run the uninstallation program by typing `uninstall -s -force`.

The uninstallation program does not produce any status messages or alerts. All status information is written to the uninstallation log file which is contained in your system's temporary directory (for example, `C:\TEMP`).

iPlanet Console Basics

Chapter 3, “Using iPlanet Console”

Chapter 4, “Servers in iPlanet Console”

Chapter 5, “User and Group Administration”

Using iPlanet Console

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

- Starting iPlanet Console and Logging In
- A Tour of iPlanet Console
- Customizing iPlanet Console
- Administration Express

Starting iPlanet Console and Logging In

iPlanet Console is a stand-alone Java application that works in conjunction with an instance of Directory Server and an instance of Administration Server on your network. Typically, you log in to iPlanet Console using your own user name and password. If the instance of Administration Server that you're logging in to requires client authentication, you will be prompted to present a client certificate. This certificate is used to create a secure channel of communication between iPlanet Console and the instance of Administration Server.

Starting iPlanet Console

The following procedures tell you how to start iPlanet Console.

To Start iPlanet Console on UNIX Systems

In the server root, enter `startconsole [arguments]` where *arguments* are any of the optional command-line arguments listed in Table 3-1.

To Start iPlanet Console on Windows NT Systems

Click Start, and then choose Programs > iPlanet Server Program Group > iPlanet Console 5.0.

Alternatively, you can start iPlanet Console in two additional ways:

- o Double-click the “startconsole” icon in your server root.
- o Enter `startconsole [arguments]` on the command line. For *arguments*, you can specify any of the arguments listed in Table 3-1.

Table 3-1 Arguments for startconsole

Argument	What it Does
<code>-a adminURL</code>	Specifies a base URL for the instance of Administration Server that you want to log in to. For example, to log in to <code>http://eastcoast.siroe.com:987</code> , you would enter the following: <code>startconsole -a http://eastcoast.siroe.com:987</code>
<code>-f fileName</code>	Captures errors and system messages to <i>fileName</i> . For example, to capture all errors and messages to a file called <code>system.out</code> , you would enter the following: <code>startconsole -f system.out</code>
<code>-h</code>	Prints out the help message for <code>startconsole</code> .
<code>-l languageCode</code>	Specifies which language this version of iPlanet Console should use. Possible values for <i>languageCode</i> are <code>en</code> , <code>fr</code> , and <code>ja</code> . For example, to start iPlanet Console in French, you would enter the following: <code>startconsole -l fr</code>
<code>-u userID</code>	Specifies the user ID to log in to iPlanet Console with. For example, to start iPlanet Console and log in with the user ID <code>bjensen</code> , you would enter the following: <code>startconsole -u bjensen</code>
<code>-w password</code>	Specifies the password for the user entered with the <code>-u</code> argument. For example, to start iPlanet Console and log in with the user ID <code>bjensen</code> and password <code>super15243</code> , you would enter the following: <code>startconsole -u bjensen -w super15243</code>

Table 3-1 Arguments for startconsole (*Continued*)

Argument	What it Does
<code>-x extraOptions</code>	<p>Specifies that you want to use extra options.</p> <p>Possible values for <i>extraOptions</i> are <code>nowinpos</code> and <code>nologo</code>. If you specify the <code>nologo</code> option, the iPlanet Console splash screen will not be displayed. If you specify the <code>nowinpos</code> option, the iPlanet Console window will be placed in the upper left corner of the screen. To specify both options, separate them with a comma.</p> <p>For example, to start iPlanet Console in the upper left corner of the screen and without a splash screen, you would enter the following:</p> <pre>startconsole -x nologo, nowinpos</pre>

Logging In to iPlanet Console With a User Name and Password

The following procedure tells you how to log in to iPlanet Console with just a user name and password. If you are logging in to an instance of Administration Server that requires you to present a client certificate, see “Logging In to iPlanet Console Using Client Authentication,” which begins on page 46.

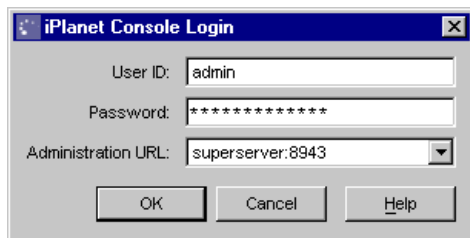
To Log in to iPlanet Console With a User Name and Password

1. Start iPlanet Console.

For more information, see “To Start iPlanet Console on UNIX Systems” on page 43 and “To Start iPlanet Console on Windows NT Systems” on page 44.

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

When specifying an Administration Server URL, you can use a hostname (such as `eastcoast.siroe.com:8943`) or IP address (such as `199.99.9.1:8943`). You do not need to include `http://` or use a fully qualified domain name, but you must include the Administration Server port number.



3. Click OK.

The user name and password you use to log in determine which servers and server operations you can access through iPlanet Console. See “Overview of Access Control” on page 171 for more information.

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

Logging In to iPlanet Console Using Client Authentication

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

The client certificates that iPlanet Console presents to an instance of Administration Server are stored in Netscape Communicator certificate database format. New and existing certificates are not recognized by Administration Server unless they are stored in the Netscape Navigator 4.7X certificate database format. For initial setup of client authentication, store certificates in the Netscape Navigator browser. After initial setup certificates can be stored in other browser certificate databases. For more information about Netscape Navigator certificate database format and certificate storage see “To Set Up Client Authentication for Users” in Chapter 10, “Using SSL and TLS with iPlanet Servers on page 211.

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

This section tells you how to do the following:

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

For more information on configuring an instance of Administration Server to require client authentication, see Chapter 10, “Using SSL and TLS with iPlanet Servers,” which begins on page 183.

To Request and Install a New Client Certificate

1. Go to the web site for a certificate authority (CA) that is trusted by the instance of Administration Server that you want to establish a secure connection with.
2. Follow the CA’s instructions to request and install a client certificate.

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

To Make Your Client Certificate Available to iPlanet Console on UNIX Systems

1. From the system prompt, go to the `.netscape` subdirectory of your home directory. For example, `/u/bjensen/.netscape`.
2. Copy the `key3.db`, `cert7.db`, and `secmodule.db` files to the `.mcc` subdirectory of your home directory.

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

To Make Your Client Certificate Available to iPlanet Console on Windows NT

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

2. Open the `Users` folder and then open your specific user folder. For example, `BJensen` (`C:\Program Files\Netscape\Users\BJensen`).
3. Copy the `key3.db`, `cert7.db`, and `secmod.db` files from your user folder to the `C:\WINNT\Profiles\your_user_ID\.mcc` folder, where `your_user_ID` is the ID that you use to log in to Windows NT.

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

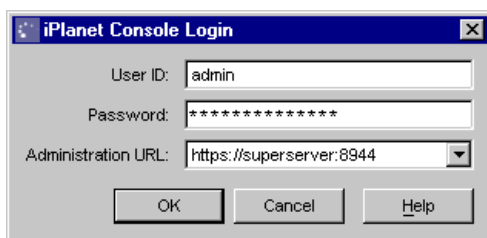
To Establish a Secure Connection With an Instance of Administration Server

1. Start iPlanet Console.

For more information, see “To Start iPlanet Console on UNIX Systems” on page 43 and “To Start iPlanet Console on Windows NT Systems” on page 44.

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

When specifying an Administration Server URL, you can use a hostname (such as `eastcoast.siroe.com:8943`) or IP address (such as `199.99.9.1:4434`). Make sure to include `https://` and the Administration Server port number in the URL.



3. Click OK.

The user name and password you use to log in determine which servers and server operations you can access through iPlanet Console. See “Overview of Access Control” on page 171 for more information.

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

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

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

A Tour of iPlanet Console

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

iPlanet Console Menus

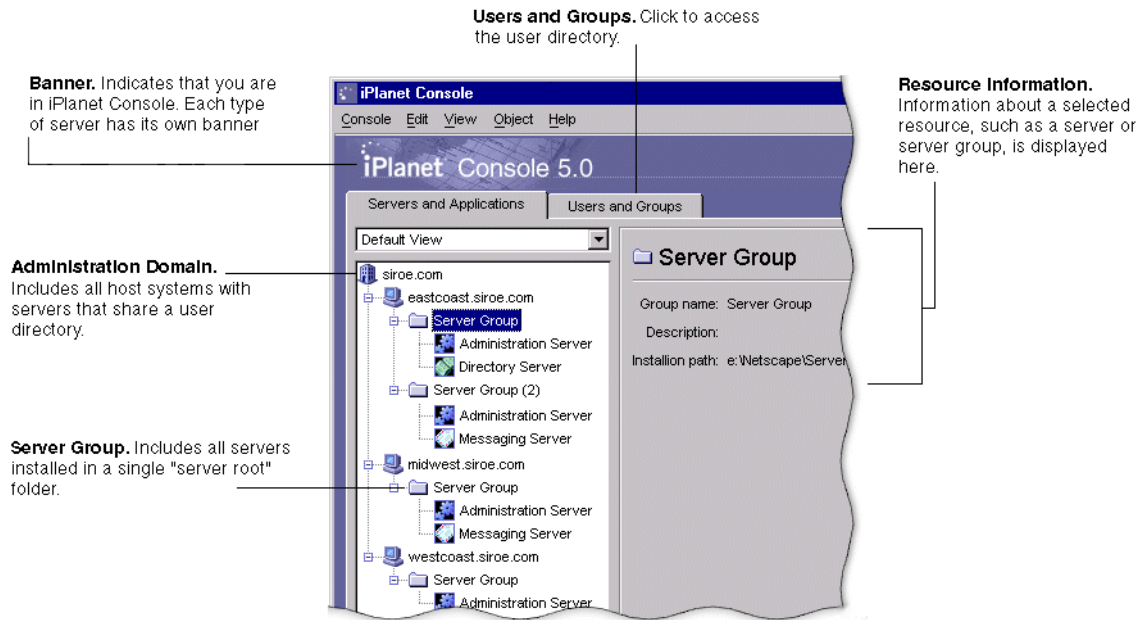
The main iPlanet Console window (shown in Figure 3-1 on page 50) has five menus: Console, Edit, View, Object, and Help. Table 3-2 summarizes what these menus are used for.

Table 3-2 iPlanet Console’s Menus and What You Can Do With Them

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

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

Figure 3-1 The Servers and Applications Tab of the Main iPlanet Console Window



iPlanet Console Tabs

The main iPlanet Console window (shown in Figure 3-1) has two tabs: “Servers and Applications” and “Users and Groups.” The “Servers and Applications” tab contains a navigation tree and an information panel. The “Users and Groups” tab has an interface that you can use to manage entries in the user directory. The “Users and Groups” tab is discussed in Chapter 5, “User and Group Administration.”

The Servers and Applications Tab

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

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

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

Figure 3-1 shows a sample navigation tree. In this example, the `siroe.com` administration domain includes three hosts. The `eastcoast` and `midwest` hosts have Messaging Server groups while the `westcoast` host contains a web server group. If the administration domain grows, an administrator can install additional server groups on these hosts. To expand a section of the navigation tree, click the plus (+) signs. To collapse a section of the tree, click the minus (-) sign.

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

For information on modifying administration domain settings, see “To Modify an Administration Domain” on page 53. For information on modifying host, server group, and instance information, see “Modifying Host, Server Group, and Instance Information” on page 77.

The Administration Domain

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

Before you can create a new administration domain, you must be a member of the Configuration Administrators group. If you are not a member of this group, you must ask your Configuration Administrator to add you to it. For instructions on adding a user to the Configuration Administrators group, see “To Add Users to the Configuration Administrators Group” on page 101.

To Create an Administration Domain

1. Open iPlanet Console.
2. From the Console menu, choose Create Administration Domain.
3. In the Create Administration Domain dialog box, enter domain information:

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

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

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

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

Directory Subtree. Enter the base DN of the user subtree in the directory. Example: `o=siroe.com`

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

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

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

4. Click OK.

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

To Modify an Administration Domain

1. In the iPlanet Console navigation tree, select the domain you want to modify, then click the Edit button in the server information panel of iPlanet Console.
2. Modify domain information as necessary:

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

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

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

```
east.siroe.com:389 west.siroe.com:393
```

See “User Authentication and Directory Failover Support” on page 130 for more information.

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

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

User Directory Subtree. Enter the base DN of the user information in the new user directory. Example: `o=siroe.com`

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

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

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

3. Click OK.

To Remove an Administration Domain

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

For more information on removing server instances, see “Removing a Server Instance” on page 78.

3. Select the administration domain that you want to remove.
4. From the Console menu, choose Remove Administration Domain.
5. Click OK.

Customizing iPlanet Console

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

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

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

Storing Display Settings

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

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

To Change Where Display Settings Are Stored

1. In iPlanet Console, from the Edit menu, choose Preferences.
2. Click the Settings tab.
3. Specify where you want to save your display settings:

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

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

4. Click OK.

To Reset Display Settings to Their Default Values

1. In iPlanet Console, from the Edit menu, choose Preferences.
2. Click the Settings tab.
3. Click the Restore Defaults button to revert to the default display settings.
4. Click OK.

Setting Display Fonts

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

To Create a Font Profile

1. In the main iPlanet Console window, from the Edit menu, choose Preferences.
2. Click the Fonts tab.
3. Click Save As, enter a name for this profile, and then click OK.

4. In the Screen Element column, click a screen element that you want to change the font for.

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

5. Click Change Font.

The Select Font dialog box appears.

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

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

Size. Choose a size for the selected font face.

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

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

Sample. This frame displays sample type using the current settings.

7. Click OK to close the Select Font dialog box.
8. If you want to set fonts for additional screen elements, repeat steps 4 through 7.
9. Click OK to save the profile.

To Edit an Existing Font Profile

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

From the Font Profile drop-down list, choose a profile. If the list is grayed out, no profiles are available.

4. Make the desired changes to the font profile.
5. Click OK to save the profile.

To Rename a Font Profile

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

From the Font Profile drop-down list, choose a profile. If the list is grayed out, no profiles are available.

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

To Use a Font Profile

1. In the main iPlanet Console window, from the Edit menu, choose Preferences.
2. Click the Fonts tab.
3. Select the font profile to use.

From the Font Profile drop-down list, choose a profile. If the list is grayed out, no profiles are available.

4. Click OK.

To Remove a Font Profile

1. In the main iPlanet Console window, from the Edit menu, choose Preferences.
2. Click the Fonts tab.
3. Select the font profile to remove.

From the Font Profile drop-down list, choose a profile. If the list is grayed out, no profiles are available.

4. Click Remove, and then confirm the deletion.
5. Click OK.

Customizing the Main Window

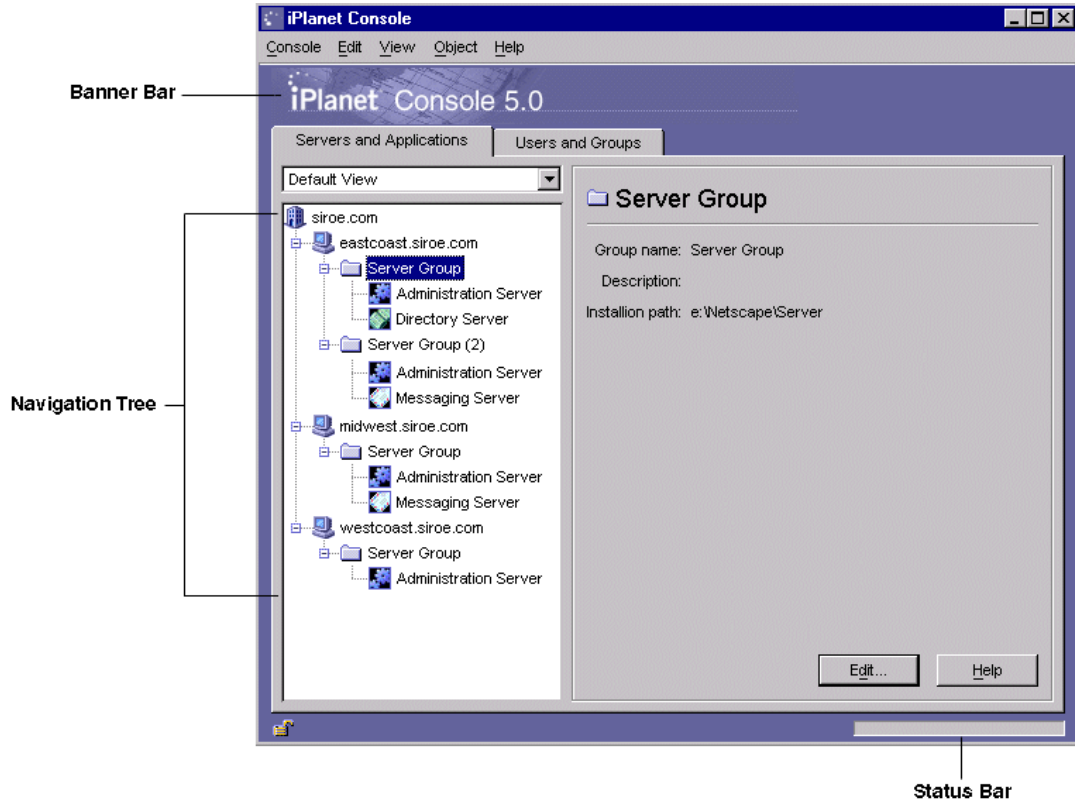
You can specify which elements of the main iPlanet Console window you want to see.

To Customize the Main Window

Select or deselect items in the View menu.

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

- Banner Bar
- Status Bar
- Tree

Figure 3-2 The Banner Bar, Navigation Tree, and Status Bar

Customizing Tables

Some iPlanet Console tasks, such as setting display fonts, use tables. You can change the position and adjust the width of columns in these tables.

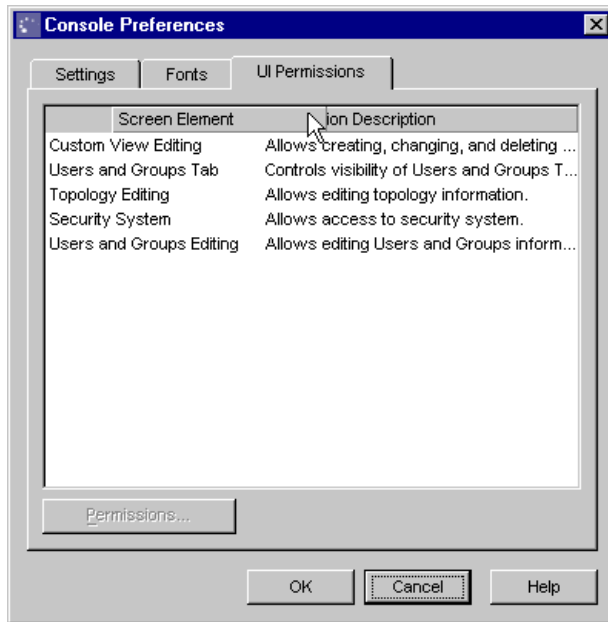
To Change Column Position in a Table

Drag each column head into the desired position.

See Figure 3-3 for an example.

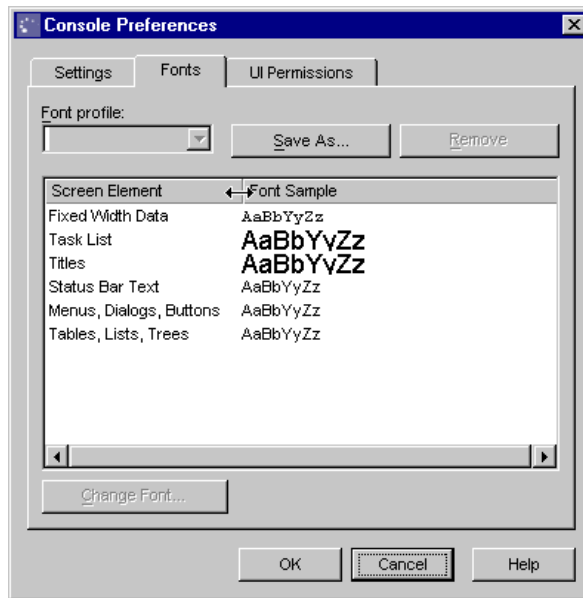
When you release the mouse button, the column will snap into its new position.

Figure 3-3 Changing the Position of a Column



To Change the Width of Columns in a Table

1. Position the pointer over a boundary of a column head.
It turns into a double arrow, as shown in Figure 3-4.
2. Drag the boundary to change the width of the column.

Figure 3-4 Resizing a Column

Creating Custom Views of the Navigation Tree

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

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

To Create a Custom View of the Navigation Tree

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

By default, a public view is visible to all users of iPlanet Console, but you can restrict access to it using access control instructions (ACIs). For more information, see “To Set Access Permissions for a Public View,” on page 65

A private view is only visible to you. You cannot apply ACIs to it.

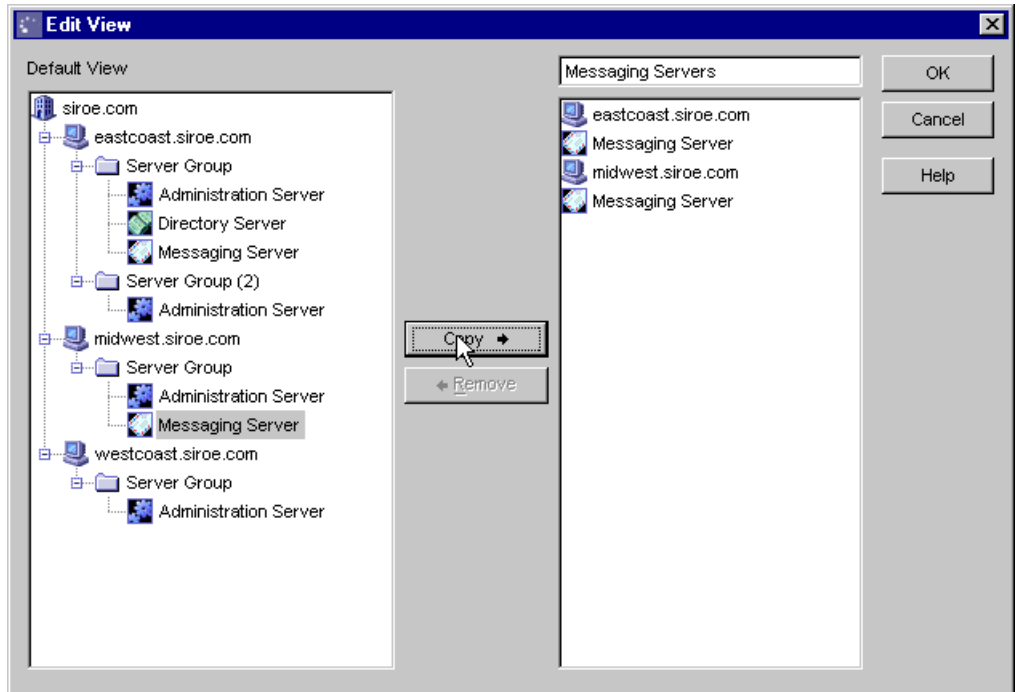
3. In the Edit View window, position your cursor in the text field and enter a descriptive name for this Custom View.
4. Select a resource from the Default View navigation tree on the left. Click Copy to include it in your Custom View navigation tree on the right.

If you need to remove a resource from the new tree, select it and click Remove.

You can select a range of resources by clicking the first item and then pressing Shift while clicking the last item. You can select multiple resources by pressing and holding down Control while clicking on one item after another.

5. Click OK when you have finished adding resources.

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

Figure 3-5 Customized Navigation Tree Example

Working With Custom Views

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

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

To Switch to a Custom View

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

Figure 3-6 Switching to a Custom View



To Edit a Custom View

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

To Rename a Custom View

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

To Set Access Permissions for a Public View

1. From the View menu, choose Custom View Configuration.
2. Choose a public Custom View from the list and click Access.
3. Specify the ACI you want to use, or create a new ACI:
 - If you want to use an existing Access Control Instruction (ACI), select it and click OK.
 - If you want to create a new ACI, click New, and then follow the directions for creating a new ACI under “Using the ACI Manager and ACI Editor” beginning on page 176.
4. Click OK when you have finished setting access permissions.

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

To Delete a Custom View

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

Administration Express

The Administration Express page is an HTML-based version of iPlanet Console that provides quick access to servers running Administration Server 4.2 or later. In the Administration Express page, you can perform four administration tasks:

- Starting servers (except stopped instances of Administration Server, which must be started from the command line)
- Stopping servers
- Viewing basic server information, such as name, description, and installation folder.
- Viewing logs

Keep the following in mind when you use the Administration Express page:

- Before you can use Administration Express to manage a server, you must upgrade its instance of Administration Server to version 4.2 or later. If you try to use Administration Express with a server using a pre-4.2 version of Administration Server, you'll get the message "Status Unknown."
- If you turn off the instance of Administration Server that you used to log in to Administration Express, you will no longer be able to use that Administration Express page. If this happens, log in again using a different Administration Server URL.

Accessing Administration Express

The Administration Express page is accessed through a browser.

To Open Administration Express

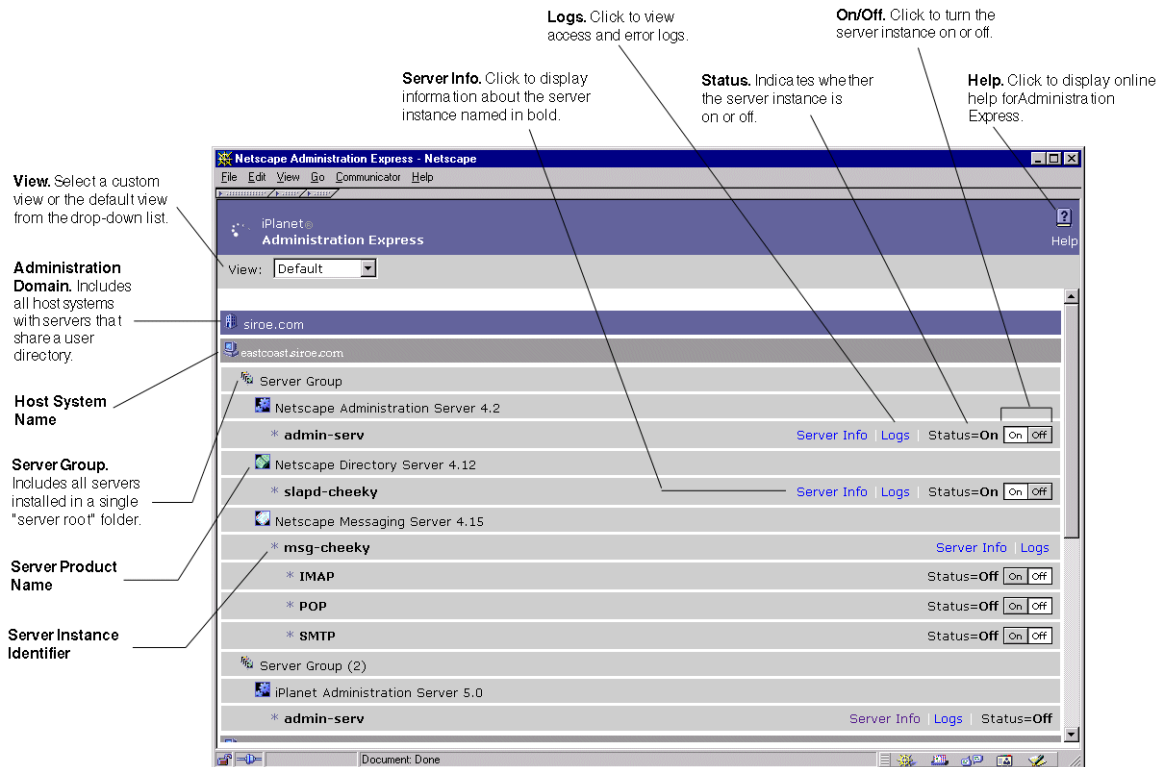
1. Open version 3.0 or later of either Netscape Navigator or Microsoft Internet Explorer, and enter the qualified host name and port number for the instance of Administration Server that you want to access.

Example: `eastcoast.siroe.com:26751`

2. In the Administration page, under Services for Administrators, click iPlanet Administration Express.
3. If prompted, enter your user name and password in the dialog box, then click OK.

If the instance of Administration Server that you are logging in to uses SSL, you may be prompted to confirm the acceptability of the instance's certificate. Additionally, if the server instance is configured to require client authentication, you may be prompted to present a client certificate. Typically, accepting server certificates involves clicking through several dialog boxes while presenting a client certificate involves making a selection from a drop-down list. If you need more information on accepting server certificates and presenting client certificates, see your browser documentation.

Once authentication is complete, you will see the main Administration Express screen:

Figure 3-7 The Administration Express Page and How to Use It

Using Administration Express

From the main Administration Express screen, you can start and stop server instances, view basic server information, and view access and error logs.

To Start or Stop a Server Instance From Administration Express

In the row containing the server instance that you want to start or stop, click On to start the server instance or Off to stop it.

Keep the following in mind when starting and stopping server instances:

- Before you can turn a server instance on or off, or view its log files, the instance of Administration Server for the server group must be running.

- You cannot use the Administration Express page to start a stopped instance of Administration Server or an instance of any server that's using SSL encryption.

UNIX

To start a stopped instance of Administration Server or an instance that's running SSL, you must always run `start-admin` from the command line. For more information on starting Administration Server, see "Restarting Administration Server." on page 113.

Windows NT

To start a stopped instance of Administration Server or an instance that's running SSL, you can run `start-admin` or use the Services control panel. For more information on starting Administration Server, see "Restarting Administration Server." on page 113.

To View Basic Server Information From Administration Express

In the row containing the server instance that you want to view information about, click Server Info.

To View Access and Error Logs From Administration Express

In the row containing the server instance that you want to view the logs for, click Logs.

Setting the Refresh Rate for Administration Express

You can configure Administration Express to automatically refresh its display of hosts and server instances. This is useful if you want to monitor the status of your iPlanet and Netscape servers and applications at regular intervals.

To Set the Refresh Rate for Administration Express

1. In a text editor, open the `serverRoot/admin-serv/config/adm.conf` file.

2. Add the following line to `adm.conf`:

```
ExpressRefreshRate: refreshRate
```

where *refreshRate* is an integer value representing the number of seconds Administration Express should wait before refreshing its display. For example, entering `ExpressRefreshRate: 120` instructs Administration Express to refresh the display every two minutes (120 seconds).

3. Save `adm.conf`.

Servers in iPlanet Console

This chapter explains how to perform basic server management using iPlanet Console. It contains the following sections:

- Working With Earlier Netscape Servers
- Working With iPlanet Servers

Working With Earlier Netscape Servers

You can use iPlanet Console to access pre-4.0 versions of Netscape servers. This section tells you how to add a pre-4.0 server to your navigation tree and how to migrate your pre-4.0 data to a newer iPlanet server.

Adding a Pre-4.0 Server to the Tree

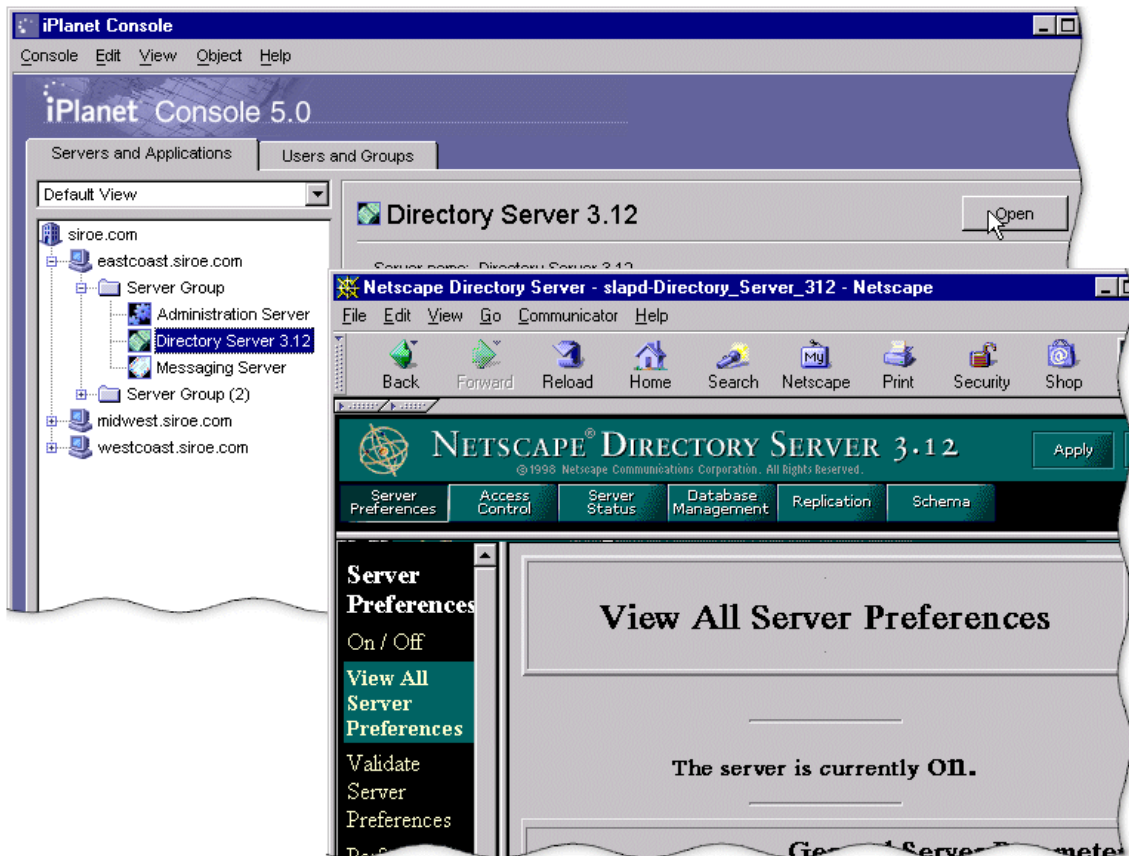
If you already have pre-4.0 versions of Netscape servers installed in your enterprise, you can access them through the iPlanet Console navigation tree. This capability is useful when you want to continue using a pre-4.0 server while preparing to deploy a newer version, and you want all servers accessible in one tree.

Pre-4.0 servers that are added to the navigation tree are not integrated completely into the iPlanet Console environment; you administer them through a browser as before. For example, you can add an existing instance of Netscape Messaging Server 3.0 to the navigation tree, but when you open that instance, the 3.0 Server Manager (which you use to administer the server) appears in a browser window.

If you want to fully integrate the information from a pre-4.0 server into iPlanet Console, you must upgrade the server to version 4.0 or later and then migrate your original configuration data to the new version. See “Migrating from a Pre-4.0 Server to a Newer Server” on page 73 for more information.

Figure 4-1 shows an example of a pre-4.0 server listed in the iPlanet Console navigation tree and managed from a browser.

Figure 4-1 A Pre-4.0 Server Listed in the Navigation Tree and Managed From a Browser



To Add a Pre-4.0 Server to the Navigation Tree

1. Open iPlanet Console and choose Add Pre-4.0 Server from the Console menu.

2. In the Add Pre-4.0 Server window, enter information for the server you want to add to the navigation tree.

Administration Server URL. Enter the host name and port number of the instance of Administration Server that you use to manage the pre-4.0 server. For example: `http://superserver.siroe.com:495`.

Server Administrator ID. Enter the user name of the administrator who manages the pre-4.0 instance of Administration Server.

Password. Enter the password for the administrator who manages the pre-4.0 instance of Administration Server.

Target Administration Domain. From the drop-down list, select the administration domain that you want to add the pre-4.0 server to.

3. Click OK.

The Server List window appears. This window lists all server instances that use the instance of Administration Server entered in step 2.

4. In the Server List window, deselect servers that you do not want to add to the navigation tree.

By default, all servers in the server root are selected for addition to the tree.

5. Click OK.

Migrating from a Pre-4.0 Server to a Newer Server

When you migrate pre-4.0 configuration settings, you copy them to a 4.0 or later server installed in a different server root. The old and new servers can co-exist on the same host system because they are installed in different server roots.

Typically, migrating the configuration settings takes less time than manually configuring a new server. It also ensures that you maintain settings that are identical to those that worked for you with the older version.

For example, if you're already using Netscape Messaging Server version 3.0, you can install Messaging Server 4.0 in a different server root. You can then migrate the 3.0 server settings to the 4.0 server. Once you're certain that the configuration settings work in the new server environment, you can safely uninstall your pre-4.0 server.

NOTE If you use the same port number for both a pre-4.0 and a newer server, you cannot run the two servers at the same time. Before starting the newer server, turn off the pre-4.0 server. Before starting the pre-4.0 server, turn off the newer server.

To Migrate From a Pre-4.0 Server to a Newer Version

1. Stop the pre-4.0 server.
2. Install the new version of the server software. When prompted, specify a server root that is different from the pre-4.0 server root.
3. Start iPlanet Console and select the server group that contains the new server.
This group becomes the target group.
4. Make sure the target group's instance of Administration Server is turned on and that you have the access privileges you need to configure a new server.
5. From the Object menu, choose Migrate Server Config.
6. In the Migrate Server Configuration window, enter the absolute path to the pre-4.0 server root folder, and then click OK.
7. In the Select Server for Migration window, check the pre-4.0 server that you want to migrate to a newer version, and then click Migrate.
8. In the "Migrate Key and Certificate" window, do one of the following:
 - If the pre-4.0 server uses SSL, provide the key password you used when you installed the server's SSL certificate, then click Migrate.
 - If the pre-4.0 server does not use SSL, click Cancel.
9. Restart the target group's instance of Administration Server.

Working With iPlanet Servers

You can perform a number of basic server tasks with iPlanet Console. This section contains the following procedures:

- Opening a server management window
- Creating a new server instance
- Cloning an iPlanet server
- Removing an iPlanet server instance
- Uninstalling an iPlanet server

Opening a Server Management Window

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

To Open an iPlanet Server Management Window

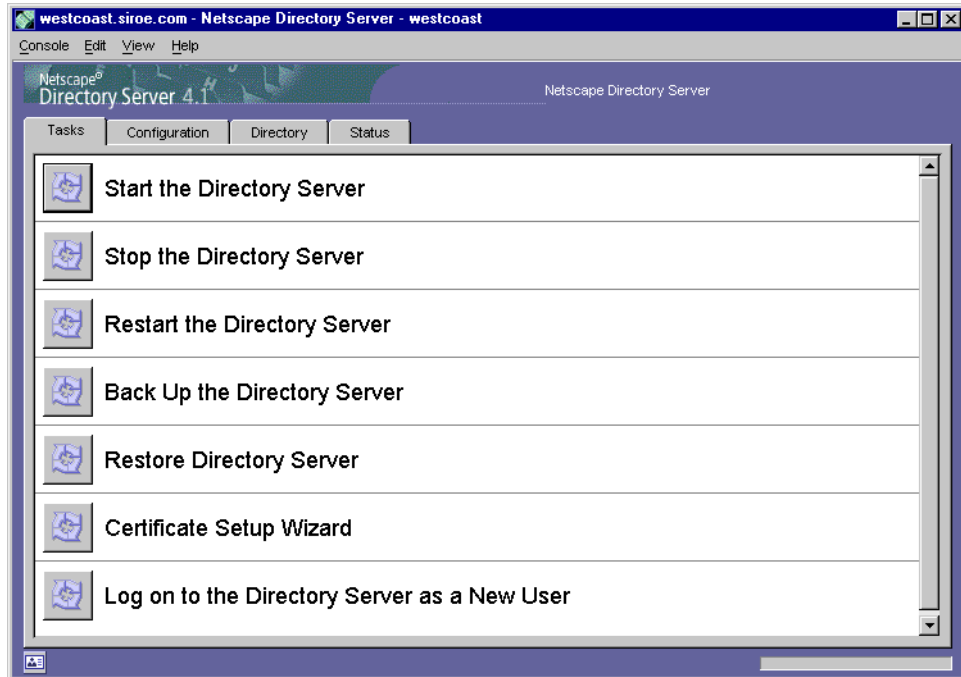
1. In iPlanet Console, click the “Servers and Applications” tab to see the navigation tree on the left and server information on the right.
2. In the navigation tree, click a server to select it.
3. In the information panel on the right side of the window, click Open.

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

Each iPlanet or Netscape server has specialized tabs for such functions as setting configurations and viewing server-specific information. For detailed information about a specific tab, see your server’s documentation. You can view many of the guides about specific products online at <http://docs.ipplanet.com>.

Figure 4-2 is an example of a server management window.

Figure 4-2 Server Management Window



Creating a New Server Instance

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

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

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

To Create a New Server Instance

1. In iPlanet 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, select the server that you want to create a new instance of.
4. Click OK.

Modifying Host, Server Group, and Instance Information

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

To Modify Host, Server Group, and Instance Information

1. In the iPlanet Console navigation tree, select the host, server group, or instance for which you want to modify information.
2. In the information panel, click Edit.
3. Edit information for the following fields:

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

- Midwest ES10000
- East Coast Sales Servers
- West Coast Messaging Server No. 3 (P-Z).

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

- Midwestern team's Sun ES10000.

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

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

4. Click OK.

Cloning a Server

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

To Clone Server Settings to Another Server

1. In the iPlanet Console navigation tree, select a reference server, the server that has the settings you want to replicate on other servers of the same type.
2. From the Object menu, choose Clone Server.
3. In the Select Target Servers for Cloning window, select the servers that you want to copy the reference server's settings to.
4. Click OK.

Removing a Server Instance

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

To Remove a Server Instance

1. In the navigation tree, select the server instance you want to remove.
2. From the Object menu, choose Remove Server.

Uninstalling an iPlanet Server

If you no longer want to create or use any instances of any particular server, you can uninstall the server. This is different from removing a server instance since all program files will be deleted. For more information on uninstallation, see “Uninstallation”, on page 37

Merging Configuration Data From Two Directory Servers

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

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

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

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

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

Figure 4-3 shows what two configuration directories might contain before they are merged.

Figure 4-3 Two Configuration Directories and the Servers They Have Settings For, Before Using the Merge Configuration Directory Utility

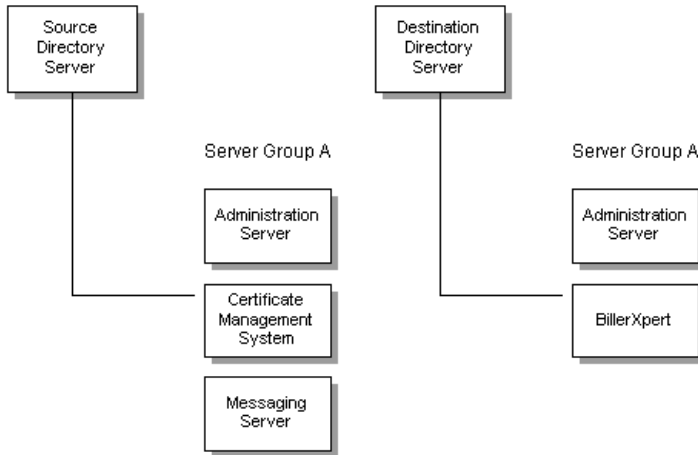
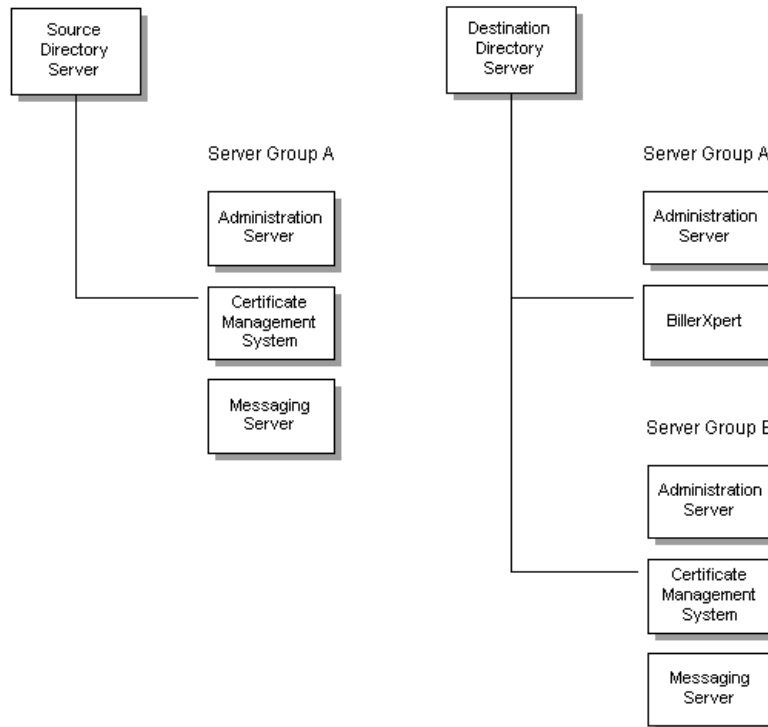


Figure 4-4 shows what the same two configuration directories would contain after you merged them.

Figure 4-4 Two Configuration Directories and the Servers They Have Settings For, After Using the Merge Configuration Directory Utility



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

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

To Merge Configuration Data From Two Directory Servers

1. In the navigation tree, select the server group containing the source configuration directory.
2. From the Object menu, choose Merge Configuration.
3. In the Merge Configuration Directory Server Information window, enter information about the configuration directory into which you want to merge the source data:

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

Destination LDAP Host. Enter the hostname for the configuration directory you specified above. Example: `eastcoast.siroe.com`

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

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

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

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

After you merge the configuration directories, the affected server instances 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 Host or Port Number”, on page 127 for more information.

User and Group Administration

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

This chapter contains the following sections:

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

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

Interacting with Directory Server

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

Using Distinguished Names

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

- Create or modify directory entries

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

From the iPlanet Console “Users and Groups” tab, you can create, select, and use directory entries.

Distinguished Names, Attributes, and Syntax

This section presents a brief summary of distinguished names, directory attributes, and syntax information. For a more detailed discussion of these concepts, see the *iPlanet Directory Server Administrator’s Guide*.

Distinguished Names

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

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

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

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

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

```
cn=Ben Hurst, ou=Operations, o=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 or organizational units (`ou`) and for the same company or organization (`o`), Klondike Corp. The third user works in a different state (`st`) from the first two users.

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

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

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

Table 5-1 Common RDN Keywords Used in DNs

RDN Keyword	Meaning in a DN	Description
c	country	Country in which the user or group resides. Examples: c=US c=GB
cn	common name or full name	Full name of person or object defined by the entry. Examples: cn=Wally Henderson cn=Database Administrators cn=printer 3b
dc	domain component	Part of a DNS domain. This keyword is typically used at the top levels of a directory tree. For example, a user in the <code>ldap.siroe.com</code> domain might have the following DN: cn=Barbara Jones,ou=Engineering,dc=siroe,dc=com
l	locality	Locality in which the user or group resides. This can be the name of a city, country, township, or other geographic regions. Examples: l=Tucson l=Pacific Northwest l=Anoka County

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

RDN Keyword	Meaning in a DN	Description
o	organization	Organization to which the user or group belongs. Examples: o=iPlanet E-Commerce Solutions o=Public Power & Gas
ou	organizational unit	Unit within an organization. Examples: ou=Sales ou=Manufacturing
sn	surname	User's last name. Example: sn=Henderson
st	state or province	State or province in which the user or group resides. Examples: st=Iowa st=British Columbia

Keep in mind that the DNs you specify when using iPlanet Console must reflect the types of data in your user directory. For information on setting up the user data in your iPlanet Directory Server see the Directory Server documentation at <http://docs.iplanet.com>.

Attributes

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

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

Table 5-2 Common User and Group Directory Attributes

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

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

Attribute Keyword	Attribute Name	Description
streetAddress	street	Street number and address of user or group defined by the entry. Example: street=494 Rice Creek Terrace
telephoneNumber	telephone	User's or group's telephone number. Example: (545) 555-1221
title	title	User's job title. Examples: title=writer title=manager
uid	user ID	Name that uniquely identifies the person or object defined by the entry.
userPassword	password	A user's password.

A user entry can include many more attributes than those listed above. In addition, you can create new attributes to meet your company's needs. For more detailed information, see the *iPlanet Directory Server Administrator's Guide*.

DN and Attribute Guidelines and Syntax

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

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

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

When schema checking is turned on, attributes must match directory schema. If you are using iPlanet Directory Server and schema checking is turned on, use RDN keywords and attributes that can be recognized by the Directory Server and are allowed by the entry's object classes. If schema checking is turned off, you can use all attributes, regardless of an entry's object classes. For more information on required attributes and schema checking, see the *iPlanet Directory Server Administrator's Guide* and the *iPlanet Directory Server Schema Reference Guide*.

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

For example,

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

is not the same as

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

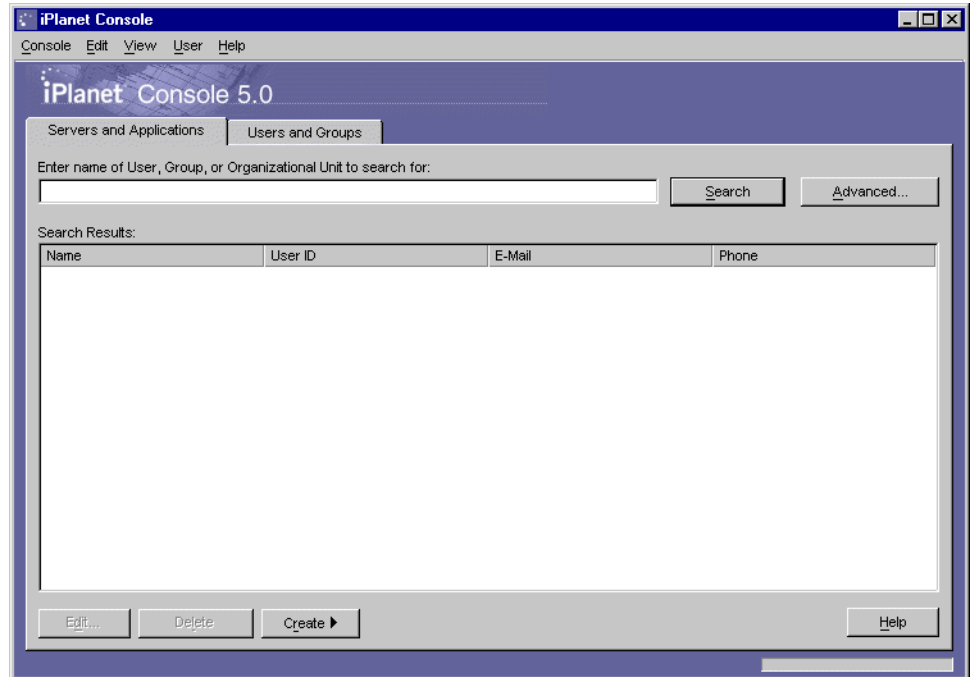
because the organizational unit (ou) and organization (o) keywords are not listed in the same order.

User IDs must be unique. If duplicate user IDs exist in your directory, users with those IDs will not be able to authenticate to the directory. Exercise caution when using the `ldapmodify` command line utility to create users, since the utility does not check for duplicate user IDs.

Locating a User or Group in the Directory

You can use the “Users and Groups” Search function to locate directory entries. Initially, the function is set to search within the default user directory. If you do not want to use the default user directory, you can manually change to another one. See “Choosing a Different Directory to Search”, on page 90 for more information.

Figure 5-1 The Users and Groups Tab of iPlanet Console



To Locate Users or Groups in the Directory

1. In iPlanet Console, click the “Users and Groups” tab.
2. Specify your search criteria in one of these ways:

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

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

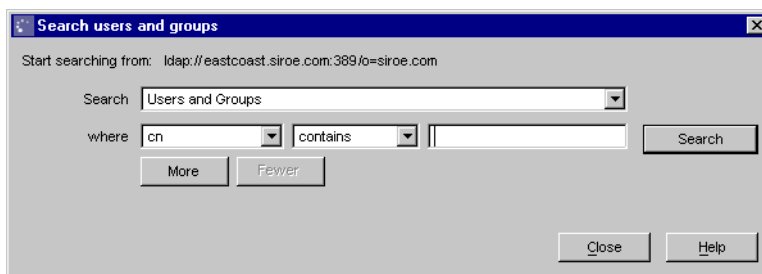
database can take a long time.

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

Search. Specify where to perform the search by choosing Users, Groups, Users and Groups, or Administrators.

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

Figure 5-2 Search Users and Groups Window



3. Click Search.

The search results are displayed in the list box.

Choosing a Different Directory to Search

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

To Change the Directory to Search

1. In iPlanet Console, click the “Users and Groups” tab.
2. From the User menu, choose Change Directory.

3. In the Change Directory dialog box, provide user directory information:
 - User Directory Host.** Enter the fully qualified host name where the user directory is installed.
 - User Directory Port.** Enter the port number used to connect to the user directory.
 - Secure Connection.** Check this box if the port number entered above is for use with the Secure Sockets Layer (SSL) protocol. Make sure that the port is configured to support SSL before selecting this option.
 - User Directory Subtree.** Enter the DN of the user directory subtree to search in. For example, to search all user entries in your organization, you might enter `o=siroe.com`. To search within the sales force, you might enter `ou=sales, o=siroe.com`.
 - Bind DN.** Enter the distinguished name of a user authorized to search entries in the user directory.
 - Bind Password.** Enter the password for the user specified by the Bind DN.
4. Click OK.

Creating New Directory Entries

From the iPlanet Console “Users and Groups” tab, you can add or modify a user, group, or organizational unit.

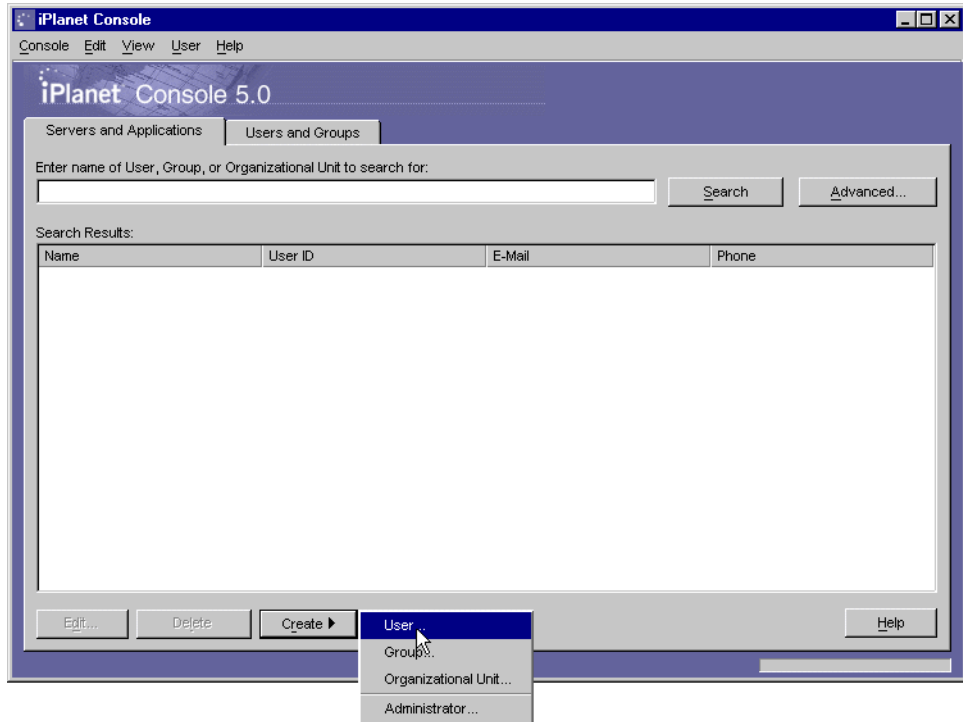
Alternatively, you can perform these directory operations from the command line. For detailed information, see the *iPlanet Directory Server Administrator’s Guide*.

Users

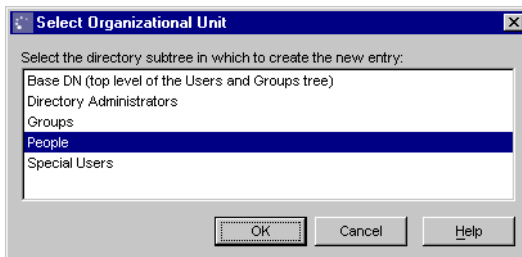
A user entry contains information about an individual person or resource in the directory. For example, you can create user entries for John Smith, Printer 3B, or Conference Room 25.

To Create a New User Entry in the Directory

1. In iPlanet Console, click the “Users and Groups” tab.
2. Click the Create button and then choose User. Alternatively you can open the User menu and choose Create > User.



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



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

First Name. Enter the user's first name.

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

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

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

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

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

E-Mail. (Optional) Enter the user's email address. If the user has multiple email addresses, separate them with commas. For example: `jd@siroe.com, john.doe@siroe.net`

Phone. (Optional) Enter the user's telephone number. If the user has multiple telephone numbers, separate them with commas. For example:
`(550)555-1212, (950)555-2121, (725)222-5151`

Fax. (Optional) Enter the user's fax number. If the user has multiple fax numbers, separate them with commas. For example: 555-2211, 555-1221

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

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

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

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

Phone. Enter the user's telephone number. If the user has multiple telephone numbers, separate them with commas. For example: (550)555-1212, (950)555-2121, (725)222-5151

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

6. If you want to specify NT- or UNIX-specific attributes, click the NT User or Posix User tab. For more information, see "Specifying Windows NT and UNIX Systems Options" on page 96.
7. Click OK.

The User's Preferred Language

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

To indicate a user's preferred language, follow the instructions in step 5 of the section "To Create a New User Entry in the Directory", beginning on page 91.

Administrators

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

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

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

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

```
<server_root>/admin-serv/config/admpw.
```

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

For more information on modifying the Configuration and Administration Server Administrators, see “Modifying Existing Directory Entries” on page 107.

To Create an Administrator

1. In iPlanet Console, click the “Users and Groups” tab.
2. Click the Create button and then choose Administrator.



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

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

The requested information is exactly the same as in the Create User dialog box, except that Password is a required field. For more information, see steps 4 through 7 of “To Create a New User Entry in the Directory” beginning on page 91.

Specifying Windows NT and UNIX Systems Options

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

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

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

To Enable Windows NT and UNIX Systems Panels for an Individual User

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

The appropriate panel appears.

2. Enable the fields in the panel.

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

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

To Enable Windows NT and UNIX Systems Panels for All New Users

1. Open your Directory Server management window.
2. Click the Directory tab and click NetscapeRoot in the navigation tree.
3. Click to open your administration domain, and then click the pluses (+) to expand GlobalPreferences > Admin > 4.0.

4. Click the defaultObjectClassesContainer folder, and then click “user” in the right-hand panel.
5. From the Object menu, choose Open.
6. Select “nsdefaultobjectclass,” then, from the Edit menu, choose Add Value.

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

7. Enter the appropriate object class name in the field.

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

8. Click OK.

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

1. Follow steps 1-5 of “To Create a New User Entry in the Directory” beginning on page 91.
2. If you want to store Windows NT-specific user information in the directory, click the NT User tab, enable the fields by selecting “Enable Windows NT user attributes,” and then enter the following information:

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

Create New NT Account. (Optional) Check this box if you are using Directory Server’s NT Synch Service and want to add this entry to the NT user database.

Delete NT Account If Person Deleted. (Optional) Check this box if you are using Directory Server’s NT Synch Service and want the delete operation to also remove this user from the NT user database. Checking this box will not delete the user. It only indicates that, if the user is deleted from the iPlanet User Directory, he will also be removed from the NT user database.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4. Click OK.

Groups

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

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

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

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

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

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

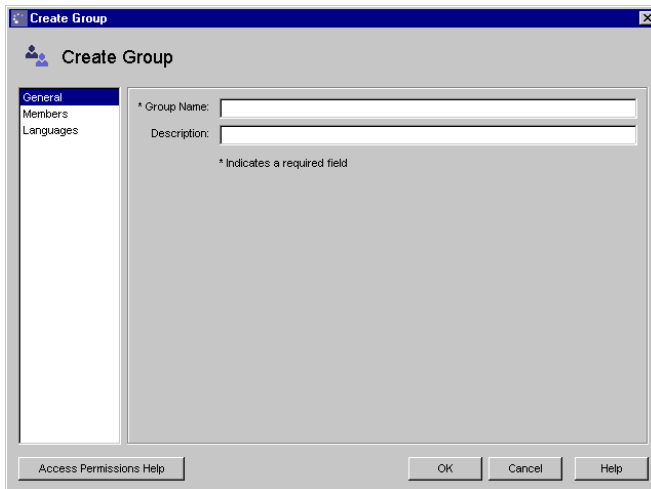
A *certificate group* includes all users who have a certificate containing a common attribute. For example, you can create a certificate group called California Western Sales whose members share these attributes: `ou=Sales`, `ou=West`, `st=CA`. When an individual user logs on to a server, if all of these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

To Create a Static Group in the Directory

1. In iPlanet Console, click the “Users and Groups” tab.
2. Click the Create button and then choose Group. Alternatively you can open the User menu and choose Create > Group.



3. In the Select Organizational Unit dialog box, select the organizational unit(ou) to which the group will belong, and then click OK.
4. In the Create Group dialog box, enter group information:
Group Name. Enter a name for the group.
Description. (Optional) Enter a description to help you identify this group.



5. Create the group, or specify members for the group before creating it.
If you want to create only the group now, and add group members later, click OK and skip the rest of this procedure.
If you want to immediately add members to the group, click Members and then continue to the next step.

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

To Add Users to the Configuration Administrators Group

1. In iPlanet Console, click the “Users and Groups” tab, and then choose Change Directory from the User menu.
2. In the Change Directory window, indicate the location of the user directory that contains the Configuration Administrators group:

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

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

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

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

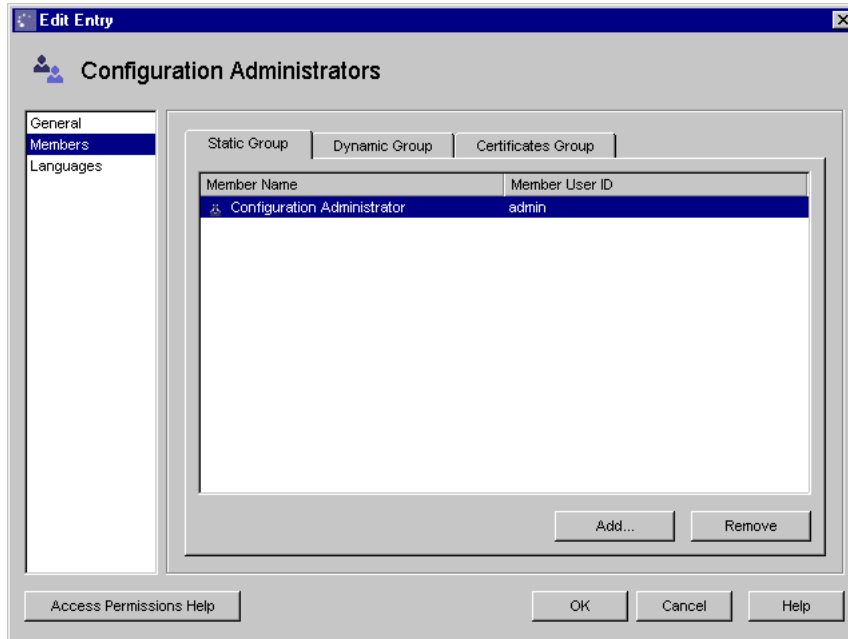
Bind Password. Enter the password of the user directory administrator.

The screenshot shows a dialog box titled "Change Directory" with the following fields and values:

- User Directory Host: eastcoast.siroe.com
- User Directory Port: 389
- Secure Connection:
- User Directory Subtree: o=NetscapeRoot
- Bind DN: uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot
- Bind Password: *****

Buttons at the bottom: OK, Cancel, Help

3. Click OK.
4. Use the Search function to locate and highlight the Configuration Administrators group, and then click Edit.
5. In the Edit Group window, click Members.



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

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

To Create a Dynamic Group

1. In iPlanet Console, click the "Users and Groups" tab.
2. Click the Create button and then choose Group. Alternatively you can open the User menu and choose Create > Group.
3. In the Select Organizational Unit dialog box, select the organizational unit (ou) to which the group will belong, and then click OK.
4. In the Create Group dialog box, enter general group information.
 - Group Name.** Enter a name for the group.
 - Description.** (Optional) Enter a description to help you identify this group.
5. Click Members.

6. Click Dynamic Group, and then click Add.
7. Use the “Construct and Test LDAP URL” dialog box to specify the criteria for including users in the dynamic group.

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

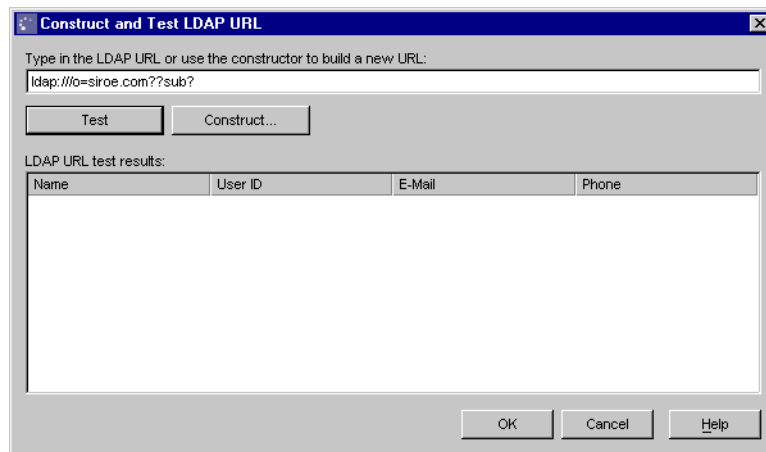
The LDAP URL will take this form:

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

For example:

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

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



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

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

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

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

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

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

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

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



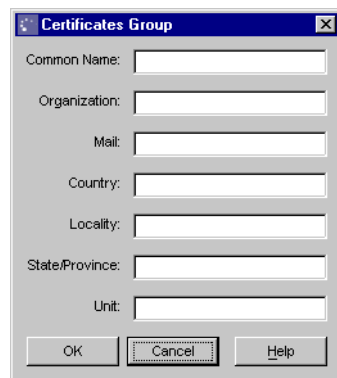
9. Click OK.
10. If you want to see a list of users and groups included in the dynamic group, click Test in the Construct and Test LDAP URL dialog box.
11. Click OK to confirm your acceptance of the LDAP URL and add it to the list used to include members in this dynamic group.

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

To Create a Certificate Group

1. In iPlanet Console, click the “Users and Groups” tab.
2. Click the Create button and then choose Group. Alternatively, you can open the User menu and choose Create > Group.
3. In the Select Organizational Unit dialog box, select the organizational unit (ou) to which the group will belong, and then click OK.

4. In the Create Group dialog box, enter group information:
Group Name. Enter a name for the group.
Description. (Optional) Enter a description that helps you identify this group.
5. Click Members
6. Click Certificate Group, and then click Add.
7. In the Certificate Group dialog box, fill in one or more of the following fields:
Common Name. Enter the full name of the group. Example: Database Administrators.
Organization. Enter the name of the organization the group belongs to. Example: Operations Group.
Mail. Enter the street address for the group.
Country. Enter the country code for the group.
Locality. Enter the city name for the group's business.
State/Province. Enter the state or province name for the group.
Unit. Enter the name of the organizational unit that the group belongs to. Example: IS Department.



The image shows a dialog box titled "Certificates Group". It contains the following fields and buttons:

- Common Name:
- Organization:
- Mail:
- Country:
- Locality:
- State/Province:
- Unit:
- Buttons: OK, Cancel, Help

8. Click OK.

Organizational Units

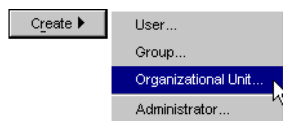
An organizational unit can include a number of groups and usually represents a division, department, or other discrete business group.

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

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

To Create a New Organizational Unit

1. In iPlanet Console, click the “Users and Groups” tab.
2. Click the Create button and then choose Organizational Unit. Alternatively, you can open the User menu and choose Create > Organizational Unit.



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

Name. Enter a name for the organizational unit.

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

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

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

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

5. Click OK.

Modifying Existing Directory Entries

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

Updating User and Group Entries

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

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

To Edit a User or Group Entry in the Directory

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

To Change a User Password

1. In the “Users and Groups” tab of iPlanet Console, use the Search function to locate the user.
2. Once the user appears in the search results list, select it, and then click Edit.
3. Enter the new password information:
Password. Enter the new password. Alphanumeric characters, spaces, and punctuation marks are all acceptable.
Confirm Password. Enter the password again to confirm.
4. Click OK for the change to take effect.

To Change the Configuration Administrator’s User Name or Password

1. In the “Users and Groups” tab of iPlanet Console, click Advanced.

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

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

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

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

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

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

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

3. Click Search.

The results appear in the “Users and Groups” tab.

4. Click Close.

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

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

First Name. Enter the administrator’s first name.

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

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

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

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

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

7. Click OK.

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

- a. Click the “Users and Groups” tab of iPlanet Console, and choose Change Directory from the User menu.
- b. In the Change Directory Window, update the Bind DN and Bind Password with the new information for the Configuration Administrator, and then click OK.

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

1. In the iPlanet Console navigation tree, select the Administration Server instance that you want to change the administrator user name or password for.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab.
4. In the Configuration tab, click the Access tab.
5. In the Access tab, enter information for the following fields:
 - Username.** Enter the user name for the Administration Server Administrator.
 - Password.** Enter the password for the Administration Server Administrator.
 - Confirm Password.** Enter the password again to confirm it.

If you make an error while entering this information, you can click Reset to restore the original values for the fields.
6. Click Save to save the new Administration Server Administrator user name or password.
7. Restart the instance of Administration Server.

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

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

If you are removing an organizational unit, you must first remove all users and groups belonging to it.
2. Click Delete.
3. Click OK when prompted to confirm the deletion.

Using iPlanet Administration Server

Chapter 6, “Administration Server Basics”

Chapter 7, “Administration Server Configuration”

Chapter 8, “Administration Server Command-Line Tools”

Administration Server Basics

iPlanet Administration Server processes requests for servers that are installed in a server group (a single root folder), and then starts the programs required to fulfill them. For a brief overview of iPlanet Console architecture, see Chapter 1, “iPlanet Console and Administration Server.”

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

- Restarting Administration Server
- Stopping Administration Server
- Logging Options
- The iPlanet Administration Page

Restarting Administration Server

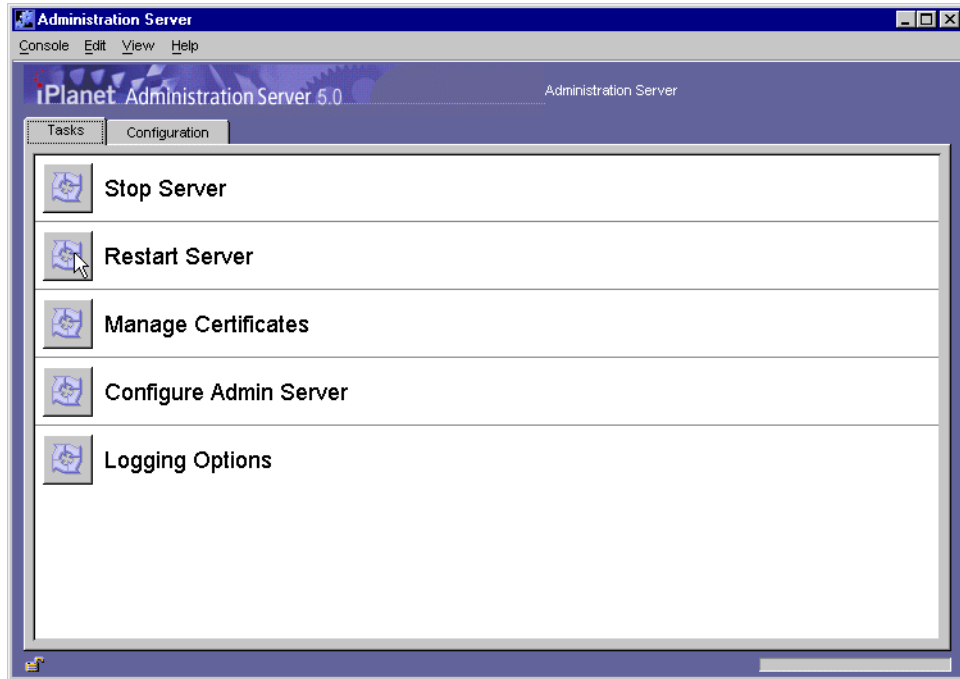
iPlanet Administration Server automatically starts once it's installed. When you need to restart Administration Server, you can do so from iPlanet Console or from the command line. In Windows NT, you can also restart the server from the Services control panel.

To Restart the Server From iPlanet Console

1. From the iPlanet Console navigation tree, select the instance of Administration Server that you want to restart.
2. Click Open to open the management window for the instance of Administration Server.

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

Figure 6-1 iPlanet Console Task Window



To Restart the Server From the Command Line

UNIX Systems

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

Windows NT Systems

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

```
serverRoot/start-admin.cmd
```

To Restart the Server From the NT Control Panel

1. Click Start, and then choose Settings > Control Panel.
2. Open the Services control panel.
3. Select iPlanet Administration Server 5.0 from the list of services and then click the Start button.
4. Click Close to exit the Services control panel.

Stopping Administration Server

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

To Stop the Server From iPlanet Console

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

To Stop the Server From the Command Line

UNIX Systems

In the server root, enter `./stop-admin`.

Windows NT Systems

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

```
serverRoot/stop-admin.cmd
```

To Stop the Server From the NT Control Panel

1. Click Start, and then choose Settings > Control Panel.
2. Open the Services control panel.
3. Select iPlanet Administration Server 5.0 from the list of services and then click Stop.
4. Click Close to exit the Services control panel.

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

Logging Options

Log files can help you monitor activity on an instance of Administration Server, and can also help you troubleshoot server problems. Server logs use the Common Logfile Format, a broadly supported format that provides information about the server.

Administration Server generates two kinds of logs:

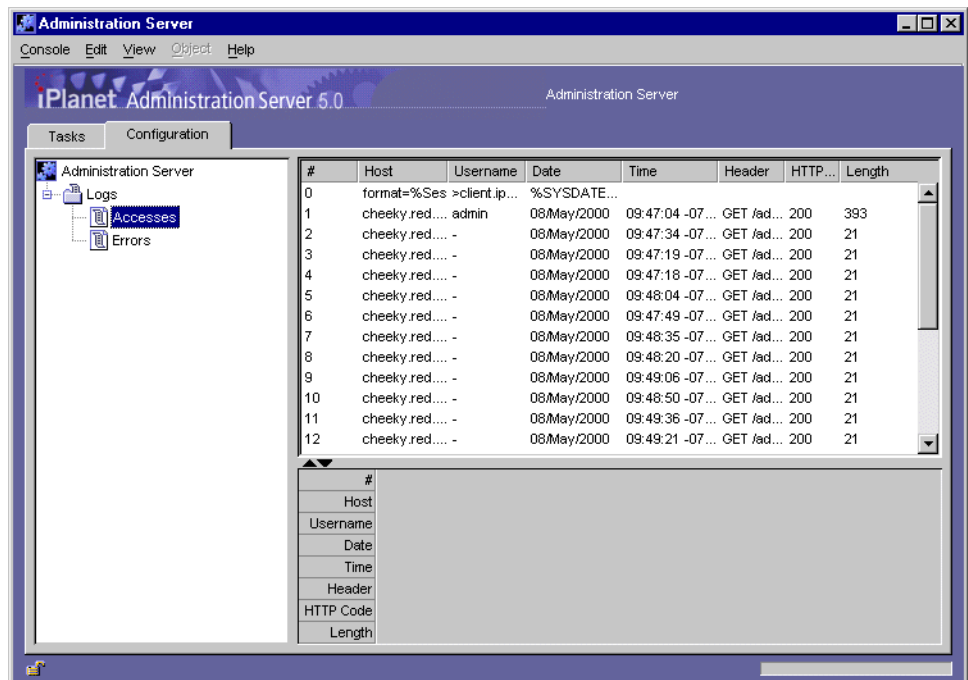
Access log. Displays information about requests to the server and the responses from the server. By default, the file is located at `admin-serv/logs/access`.

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

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

To View the Access Log

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



To View the Error Log

1. From the iPlanet Console navigation tree, select the instance of Administration Server that you want to view the error log for.
2. Click Open to open the management window for the instance of Administration Server.

3. Click the Configuration tab.
4. In the configuration tree, click + to expand the Logs directory, then click the Errors icon.

If you want to resize the column widths to show more detail, move your mouse to position the pointer over a column head boundary so that it changes to a double-arrow. Then, drag to make the column the width you want.

To Change Where Logs Are Stored

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

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

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

5. Click OK.

The iPlanet Administration Page

The iPlanet administration page provides links to sites or services of interest to system administrators. For example, in , the administration page contains a link to Administration Express and a link to a web site for downloading server software. Depending on which iPlanet server products you have installed, the administration page may include links to additional resources, such as the Netscape Directory Server gateway or the Netscape Directory Server end-user pages.

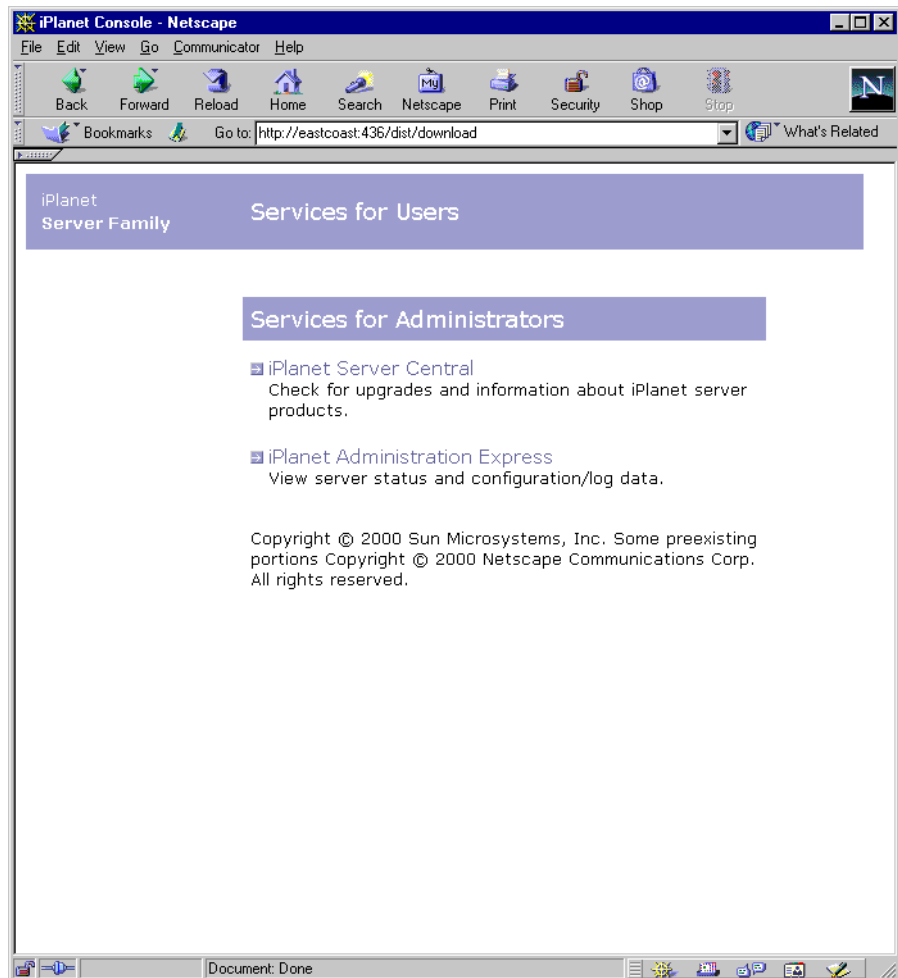
For more information on Administration Express, see “Administration Express” on page 65. For more information on links related to other Netscape or iPlanet server software, see your server’s documentation.

To Access the Administration Page

1. Open a browser.
2. Enter the fully qualified host name and port number for the instance of Administration Server you want to access.

Example: `http://eastcoast.siroe.com:26751`

3. Press Enter.



Administration Server Configuration

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

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

Network Settings

Network settings affect the way an instance of iPlanet Administration Server runs. By default, these settings are configured automatically during installation, but you can modify them if your system configuration changes. You can change the following settings:

- Port Number
- Connection Restrictions

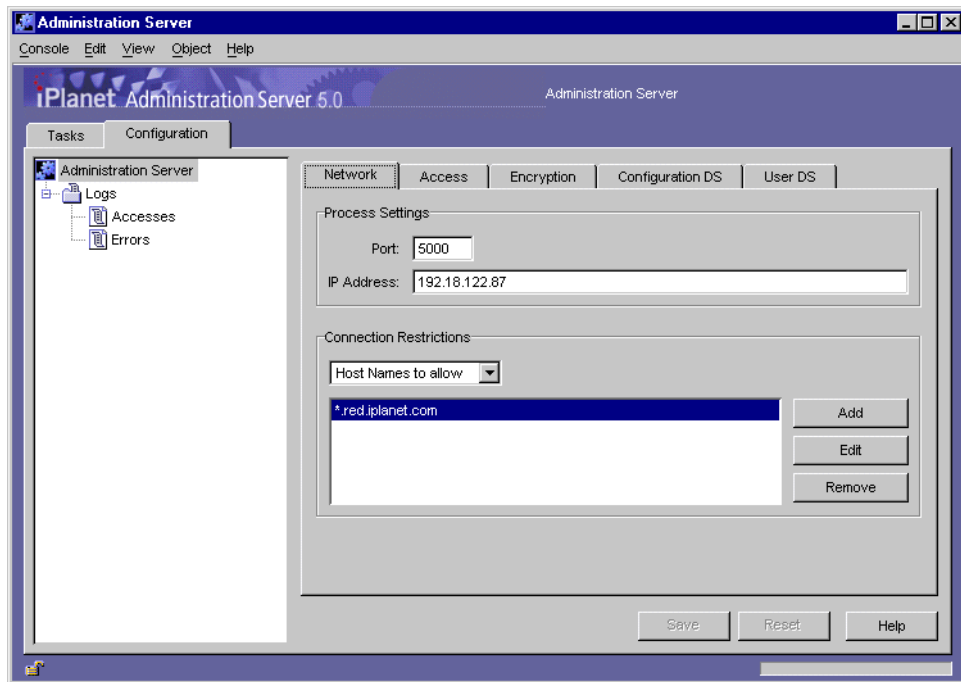
The *port number* specifies where an instance of Administration Server listens for messages. It can be any number between 1 and 65535 but, to avoid conflicts with other resources, it is typically a number greater than 1024. For security reasons, consider changing the port number regularly.

Connection restrictions allow you to specify which hosts are allowed to connect to an instance of Administration Server. You can list these hosts by DNS name, IP address, or both. You can use the * wildcard to specify a group of hosts. For instance, entering *.siroe.com allows all machines in the siroe.com domain to

access the instance. Entering `205.12.*` allows all hosts whose IP addresses begin with `205.12` to access the instance. When specifying IP address restrictions, you must include all three separating dots. If you do not, you will receive an error message.

To Configure Network Settings

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



4. Enter network settings:

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

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

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

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

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

5. Click OK.

Access Settings

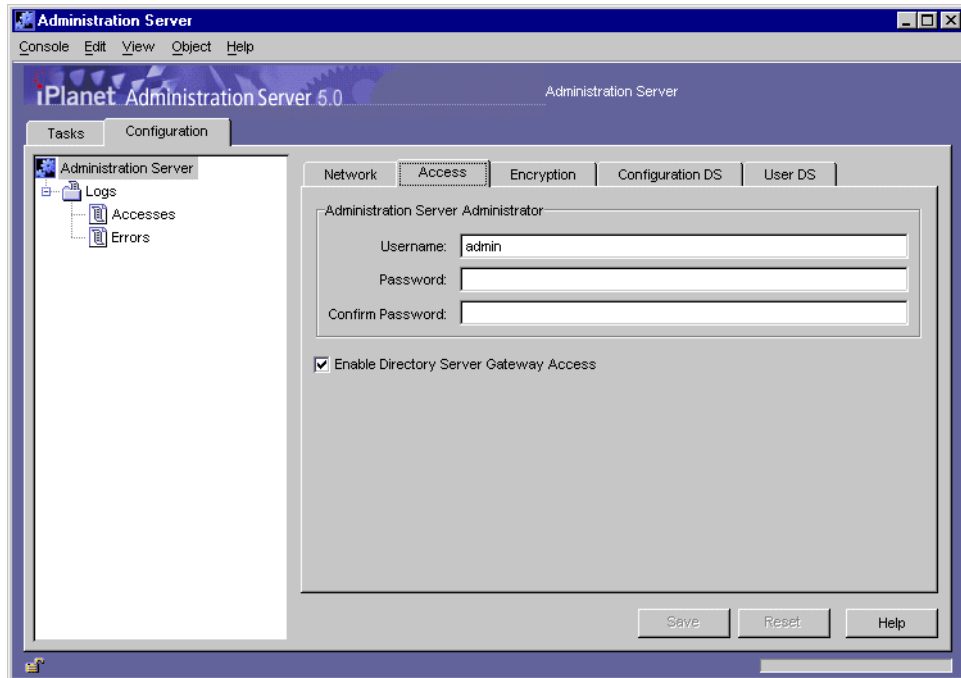
You can use the Access Settings tab to specify a user name and password for the Administration Server Administrator and to enable or disable Directory Server Gateway access.

The Administration Server Administrator is a special user that has full access to all features in an instance of Administration Server. This user is created during installation for the purpose of starting iPlanet Console if a Directory Server is unavailable. The Administration Server Administrator user name and password are stored in the file `serverRoot/admin-serv/config/admpw`.

The Directory Server Gateway is a service that provides web-based access to the entire user directory. The Directory Server Gateway must be installed before you can use this option. See the *iPlanet Directory Server Administrator's Guide* for more information.

To Set Administration Server Access Settings

1. From the iPlanet Console navigation tree, select the instance of Administration Server that you want to set Access Settings for.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab, and then click the Access tab.



4. Enter access information:
 - User name.** Enter the user ID for the Administration Server Administrator.
 - Password.** Enter the Administration Server Administrator's password.
 - Confirm Password.** Enter the password again to confirm it.
 - Enable Directory Server Gateway Access.** By default, this option is selected for you. Deselect it to disable access to the Directory Server gateway.
5. Click OK.

Encryption Settings

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

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

To Request and Install a Certificate for Administration Server

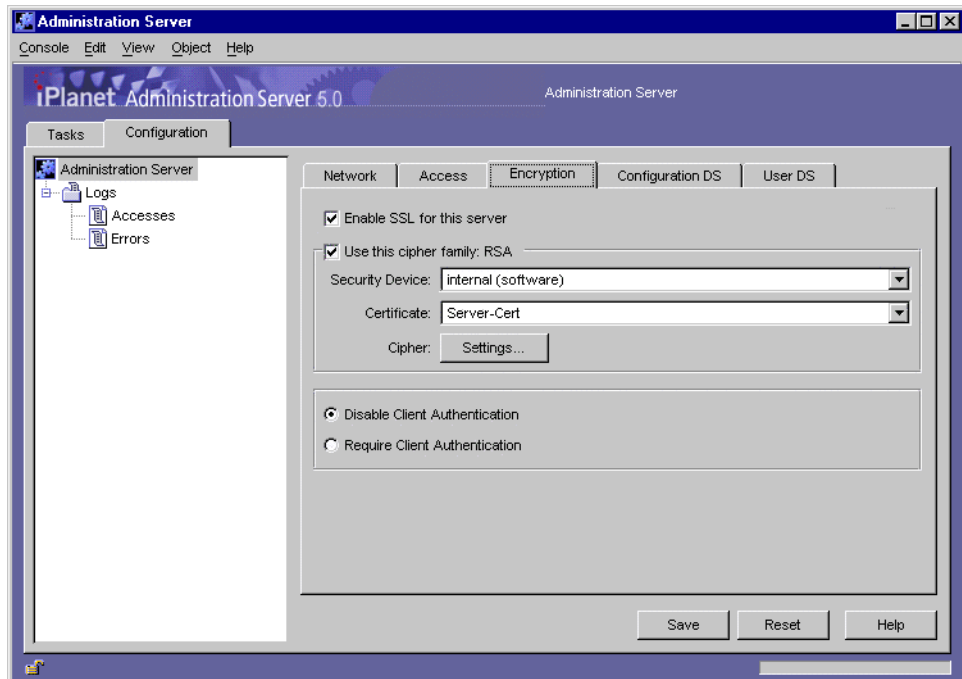
1. In the iPlanet Console navigation tree, select the instance of Administration Server that you want to install a certificate on.
2. Click Open to open the management window for the instance of Administration Server.
3. In the Administration Server management window, open the Console menu, and choose Security > Manage Certificates.
4. Click the Request button, and then provide information as prompted.
See “Obtaining and Installing a Server Certificate” on page 187 for detailed information.
5. Once you have a certificate, click the Install button, and then provide information as prompted.
See “Obtaining and Installing a Server Certificate” on page 187 for detailed information.

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

To Activate SSL on Administration Server

1. In the iPlanet Console navigation tree, select the instance of Administration Server that you want to activate SSL encryption on.

2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab.
4. Click the Encryption tab.



5. Select "Enable SSL for this server."

The options in the following steps become available only when you turn on SSL encryption.

6. Select "Use this cipher family: RSA."
7. Choose the security device where your key is stored:

If the key is stored in the local key database, select "Internal (Software-based)."
If the key is stored on an external device (such as a SmartCard), select that device.

8. Choose the certificate you want to use with SSL.

Certificate information is stored in the certificate database. If you're not sure which certificate to use, view the Certificate Management dialog box for more information. To view the Certificate Management dialog box, from the File menu, choose Certificate Management.

9. Click the Settings button.
10. Set the ciphers that this instance of Administration Server should accept when communicating securely with iPlanet Console, other servers, or browsers.

First, click a tab for a version of SSL or TLS. Then, choose the ciphers that you want this instance of Administration Server to accept when communicating over that version of SSL or TLS.
11. Click Save.

Directory Settings

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

The Configuration Directory

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

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

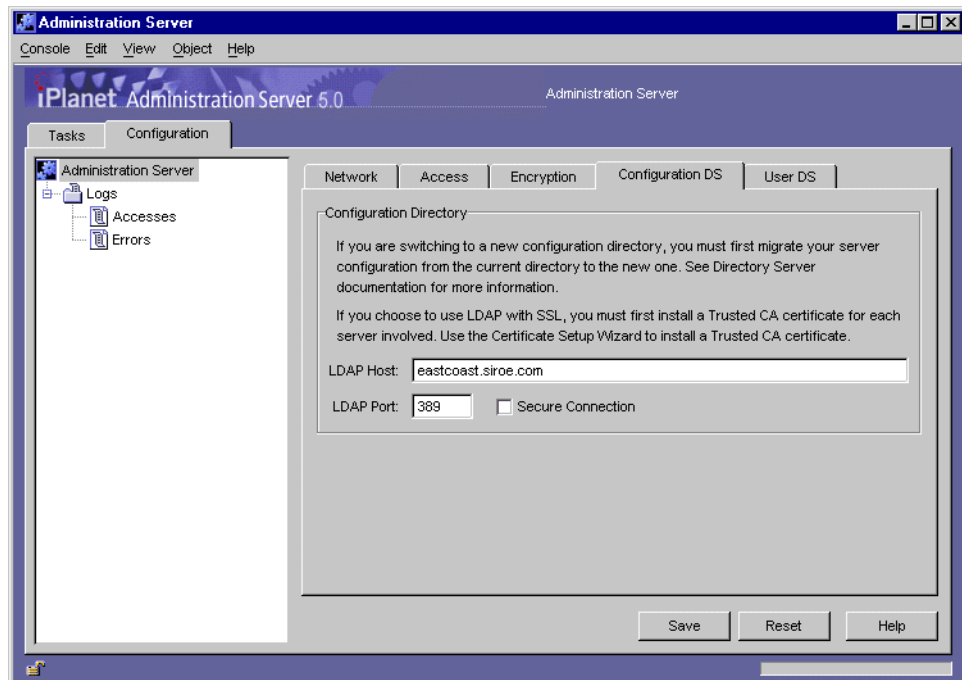
Changing the Host or Port Number

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

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

To Change the Host or Port Number

1. In the iPlanet Console navigation tree, select the instance of Administration Server that you want to change configuration Directory Server settings for.
2. Click Open to open the management window for the instance of Administration Server.
3. Click the Configuration tab.
4. Click the Configuration DS tab.



5. Modify settings as appropriate:

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

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

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

6. Click Save.

The User Directory

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

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

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

User Directory Settings

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

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

User Authentication and Directory Failover Support

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

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

To list the user directories to use for failover support, follow the instructions for "Changing User Directory Settings for a Domain" on page 130 or "To Change User Directory Settings for a Server Group." on page 132. For information on replicating the user directory, see the *Directory Server Administrator's Guide*.

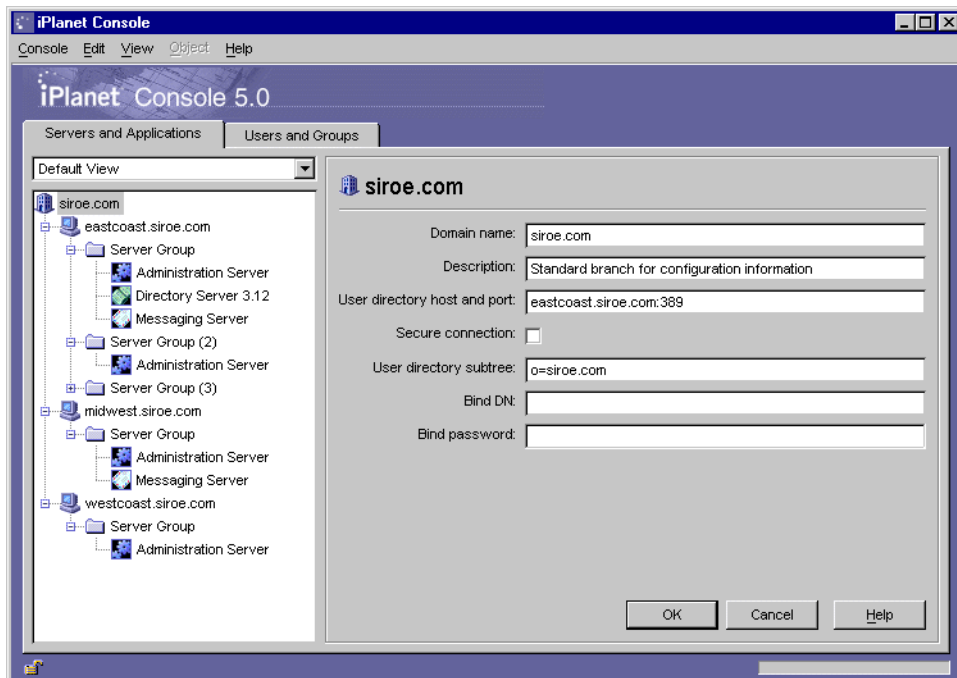
Changing User Directory Settings for a Domain

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

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

To Change the User Directory Settings for a Domain

1. In the iPlanet Console navigation tree, select the administration domain that you want to change user directory settings for.
2. In the right-hand panel of the main iPlanet Console window, click Edit.



3. Modify domain information as appropriate.

Domain name. Enter a domain name. Example: `eastcoast.siroe.com`

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

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

```
eastcoast.siroe.com:389 westcoast.siroe.com:4332
```

See "User Authentication and Directory Failover Support" on page 130 for more information.

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

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

User directory subtree. Enter the location of the new user directory. Example:
`o=siroe.com`

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

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

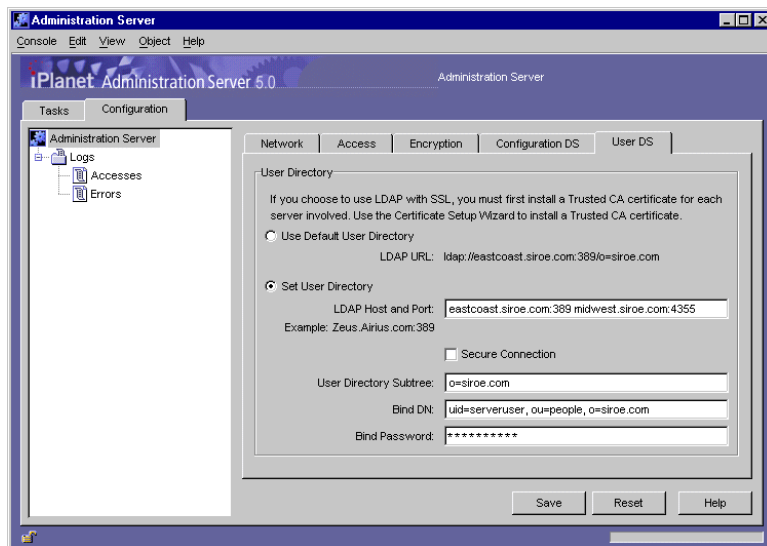
Bind password. (Optional) Enter the password of the user specified by the Bind DN.

4. Click Save.

To Change User Directory Settings for a Server Group

1. In the iPlanet Console navigation tree, click + to expand the server group that you want to change user directory settings for.
2. Select the instance of Administration Server in the server group.
3. Click Open to open the management window for the instance of Administration Server.
4. Click the Configuration tab.

5. Click the User DS tab.



6. Modify settings as appropriate.

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

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

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

Example:

```
eastcoast.siroe.com:389 westcoast.siroe.com:4332
```

See "User Authentication and Directory Failover Support" on page 130 for more information.

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

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

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

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

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

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

7. Click Save.

Administration Server Command-Line Tools

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

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

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

admconfig

The `admconfig` utility allows you to configure an instance of Administration Server using the command line instead of using the iPlanet Console graphical interface. Use `admconfig` to modify network, access, encryption, or directory settings. The utility is stored at `serverRoot/bin/admin`.

Syntax

```
admconfig [options] task [args] [task2] [args] [task3] [args] ...
```

The options that you can use with `admconfig` are described in the section that follows. The tasks that you can perform with `admconfig`, as well as the arguments for those tasks, are described in “Tasks and Their Arguments,” which begins on page 137.

Options

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

Option commands are not case sensitive. For example, both `-USER` and `-User` are accepted as the `-user` command. You can use multiple option commands with the same invocation of `admconfig`. For example, the following option commands specify that `admconfig` should establish an encrypted connection with `eastcoast.siroe.com` on port 904.

```
-enc -ser eastcoast.siroe.com:904
```

Table 8-1 Options You Can Use With `admconfig`

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

Table 8-1 Options You Can Use With admconfig (Continued)

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

Tasks and Their Arguments

A task specifies an operation that `admconfig` should perform. Some tasks take arguments, commands that provide information necessary to complete an operation.

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

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

Table 8-2 Tasks You Can Perform With admconfig

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

Table 8-2 Tasks You Can Perform With admconfig (Continued)

Commands for Tasks	What the Command Does
<code>-viewA[ccessLogEntries]</code>	<p>Lets you view the specified entries in the error log file.</p> <p>Syntax</p> <pre>admconfig [options] -viewAccessLogEntries \"start stop\"</pre> <p>Required Arguments</p> <p><i>start</i> The number of the first log entry to display.</p> <p><i>stop</i> The number of the last log entry to display.</p> <p>On UNIX systems, the backslash character is required before the quotes surrounding the <i>start</i> and <i>stop</i> arguments. If the backslash is not provided, the shell will evaluate the quotes and pass the arguments without quotes to the command line. As a result, only <i>start</i> will be assigned as a parameter for <code>-viewAccessLogEntries</code>, causing the operation to fail.</p>
<code>-countE[rrorLogEntries]</code>	<p>Counts the number of entries in the error log file. Run this task prior to <code>-viewErrorLogEntries</code> to determine the number of entries in the error log.</p>
<code>-viewE[rrorLogEntries]</code>	<p>Lets you view the specified entries in the error log file.</p> <p>Syntax</p> <pre>admconfig [options] -viewErrorLogEntries \"start stop\"</pre> <p>Required Arguments</p> <p><i>start</i> The number of the first log entry to display.</p> <p><i>stop</i> The number of the last log entry to display.</p> <p>On UNIX systems, the backslash character is required before the quotes surrounding the <i>start</i> and <i>stop</i> arguments. If the backslash is not provided, the shell will evaluate the quotes and pass the arguments without quotes to the command line. As a result, only <i>start</i> will be assigned as a parameter for <code>-viewErrorLogEntries</code>, causing the operation to fail.</p>
<code>-enableD[SGWAccess]</code>	<p>Enables access to this instance of Administration Server from the Directory Server gateway.</p>
<code>-disabledE[SGWAccess]</code>	<p>Disables access to this instance of Administration Server from the Directory Server gateway.</p>

Table 8-2 Tasks You Can Perform With admconfig (Continued)

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

Table 8-2 Tasks You Can Perform With admconfig (Continued)

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

Table 8-2 Tasks You Can Perform With admconfig (Continued)

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

Table 8-2 Tasks You Can Perform With admconfig (Continued)

Commands for Tasks	What the Command Does
-setUGDSConfig]	<p>Specifies the host, port, base DN, authentication DN, and authentication password for the instance of Directory Server containing the user and group directory.</p> <p>You can invoke <code>-setUGDSConfig</code> either with or without arguments. If you invoke this task without any arguments, the Directory Server configuration is reset to the installation defaults.</p> <p>Syntax</p> <pre>admconfig [options] -setUGDSConfig ["host port baseDN ssl uid pwd"]</pre> <p>Optional Arguments</p> <p>If you want to override the current user and group settings, you must provide all six of the following arguments:</p> <ul style="list-style-type: none"> • <i>host</i> The host name on which the instance of Directory Server is running. • <i>port</i> The port number on which the instance of Directory Server is running. • <i>baseDN</i> The base DN for the instance of Directory Server. • <i>ssl</i> Specify <code>true</code> or <code>false</code> depending on whether the instance of Directory Server is already using the Secure Sockets Layer (SSL) protocol to communicate with this instance of Administration Server. • <i>uid</i> The Distinguished Name used to bind to the instance of Directory Server. Example: <code>dn: uid=scarter, ou=people, dc=siroe, dc=com</code> • <i>pwd</i> The password used to bind to the instance of Directory Server. <p>On UNIX systems, the backslash character is required before the quotes surrounding these arguments. If the backslash is not provided, the shell will evaluate the quotes and pass the arguments without quotes to the command line. As a result, only <i>host</i> will be assigned as a parameter for <code>-setUGDSConfig</code>, causing the operation to fail.</p> <p>The <i>host</i>, <i>port</i>, <i>baseDN</i>, and <i>ssl</i> arguments are used to create the LDAP URL for the <code>ugdsconfig.dirurl</code> attribute. The <i>uid</i> argument is used to set the <code>ugdsconfig.binddn</code> attribute, and the <i>pwd</i> argument is used to set the <code>ugdsconfig.bindpw</code> attribute.</p>

Table 8-2 Tasks You Can Perform With admconfig (Continued)

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

Table 8-2 Tasks You Can Perform With admconfig (Continued)

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

Examples

The following examples demonstrate different uses of `admconfig`.

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

```
admconfig -server eastcoast.siroe.com:22222 -user john:password
-verbose 5 -setPort 33333 -restart
```

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

```
admconfig -ser eastcoast.siroe.com:33333 -u john:password -verb
-geth
```

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

```
admconfig -h r
```

admin_ip.pl

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

A Perl script is provided to help you update these two configurations. The script changes the IP address for an instance of Administration Server in both the `local.conf` file and the configuration directory. The script is called `admin_ip.pl` and is stored in the `serverRoot/shared/bin` folder.

Usage

To run `admin_ip.pl` follow the instructions for UNIX Systems or Windows NT as appropriate:

On UNIX Systems

In the `serverRoot/shared/bin` folder, enter

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

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

On Windows NT

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

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

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

ldapsearch, ldapmodify, and ldapdelete

These tools allow you to search and modify the user directory. They are stored in the `serverRoot/shared/bin` folder. For detailed information about how to use these tools, see the *iPlanet Directory Server Administrator's Guide*.

sec-activate

The `sec-activate` tool is used to activate and deactivate SSL for an instance of Administration Server. The `sec-activate` program is stored in the `serverRoot/bin/admin/admin/bin` folder.

Syntax

```
sec-activate serverRoot SSLEnabled
```

Enter information for the following variables:

serverRoot. The server root of the instance of Administration Server on which you want to activate or deactivate SSL.

SSLEnabled. Either `on` or `off`.

Example

```
sec-activate /usr/iplanet/server4 off
```

sec-migrate

The `sec-migrate` tool migrates keys and certificates from a pre-4.0 Netscape server to a target iPlanet or Netscape 4.x server. Migrating keys and certificates is useful when you want to use a pre-4.0 SSL certificate with a new server. This tool allows you to use the existing pre-4.0 certificate and its key instead of obtaining a new certificate. The `sec-migrate` program is stored in the `serverRoot/bin/admin/admin/bin` directory.

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 server root.

sie. Server instance entry: Name of the server instance to migrate key and certificate information to.

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

modutil

The `modutil` tool is a command-line utility for managing PKCS #11 module information stored in `secmod.db` files or hardware tokens. You can use the tool to perform the following operations:

- Adding and deleting PKCS #11 modules
- Changing passwords
- Setting defaults
- Listing module contents
- Enabling or disabling slots
- Enabling or disabling FIPS-140-1 compliance
- Assigning default providers for cryptographic operations
- Creating `key3.db`, `cert7.db`, and `secmod.db` security database files.

Security module database management is part of a process that typically 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.

The `modutil` tool is stored in the `serverRoot/shared/bin` folder.

Syntax

To run the `modutil` tool, enter the following command:

```
modutil task [option]
```

In this syntax *task* and *[option]* are a combination of a task and an option, from Table 8-3 and Table 8-4, each invocation of `modutil` can take one task and one option. Each option may take zero or more arguments. To view usage information, run the command without options.

Tasks and Options

You can use the `modutil` tool to perform a number of different tasks. These tasks are specified through the use of commands and options. Commands specify the task to perform. Options modify a task command.

Table 8-3 and Table 8-4 define the task commands and options for `modutil`.

The “Usage” section on page 145 section gives similar information, but by administrator task rather than by command.

Task Commands

Table 8-3 describes what the `modutil` commands do and what options are available for each. Table 8-4 defines what the options do.

Table 8-3 Task Commands and Options for `modutil`

Commands for Tasks	What the Command Does and Options for It
<code>-add moduleName</code>	<p>Adds the named PKCS #11 module to the database.</p> <p>You can use the following options with this command:</p> <ul style="list-style-type: none"> • <code>-libfile libraryFile</code>, to specify a DLL or library containing the implementation of the module • <code>-ciphers cipherList</code>, to enable specific ciphers for the module • <code>-mechanisms mechanismList</code>, to specify which security mechanisms this module will be the default service provider for

Table 8-3 Task Commands and Options for modutil (*Continued*)

<code>-changepw <i>token</i></code>	<p>Changes the password for the named token. If the token has not been initialized, this option initializes it with the supplied password. In this context, the term “password” is equivalent to a personal identification number (PIN).</p> <p>You can use the following options with this command:</p> <ul style="list-style-type: none"> • <code>-pwfile <i>passwordFile</i></code>, to specify a text file that contains the token’s current password • <code>-newpwfile <i>newPasswordFile</i></code>, to specify a text file that contains the token’s new password
<code>-create</code>	<p>Creates new <code>secmod.db</code>, <code>key3.db</code>, and <code>cert7.db</code> files.</p> <p>You can use the following option with this command:</p> <p><code>-dbdir <i>dbFolder</i></code></p> <p>If any of these security databases already exist in a specified directory, the <code>modutil</code> tool displays an error message.</p>
<code>-default <i>moduleName</i></code>	<p>Specifies the security mechanisms for which the named module will be a default provider.</p> <p>This command uses the following option:</p> <p><code>-mechanisms <i>mechanismList</i></code></p>
<code>-delete <i>moduleName</i></code>	<p>Deletes the named module.</p> <p>Note: You cannot delete the Netscape internal PKCS #11 module.</p>
<code>-disable <i>moduleName</i></code>	<p>Disables all slots on the named module.</p> <p>To disable a specific slot, use the following option:</p> <p><code>-slot <i>slotName</i></code></p>
<code>-enable <i>moduleName</i></code>	<p>Enables all slots on the named module.</p> <p>To enable a specific slot, use the following option:</p> <p><code>-slot <i>slotName</i></code></p>

Table 8-3 Task Commands and Options for modutil (*Continued*)

<code>-fips <i>true_or_false</i></code>	<p>Enables or disables FIPS-140-1 compliance for the Netscape internal module.</p> <p>To enable compliance, enter <code>-fips true</code>. To disable compliance, enter <code>-fips false</code>.</p>
<code>-force</code>	<p>Disables the <code>modutil</code> tool's interactive prompts so it can be run from a script. Use this command 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.</p>
<code>-jar <i>JARfile</i></code>	<p>Adds a new PKCS #11 module to the database. The module must be contained in the named JAR file.</p> <p>The JAR file identifies all files to install, the module name, mechanism flags, and cipher flags. It should also contain any files to be installed on the target machine, including the PKCS #11 module library and other files such as documentation.</p> <p>The JAR file uses the iPlanet Server PKCS #11 JAR format. See “JAR Information File” on page 155 for more information on creating iPlanet JAR files.</p> <p>You can use the following options with this command:</p> <p><code>-installdir <i>installationFolder</i></code>, to specify the root installation folder for the files contained in the JAR file.</p> <p><code>-tempdir <i>temporaryFolder</i></code>, to specify the folder in which to store temporary files created by the <code>-jar</code> task command</p>
<code>-list [<i>moduleName</i>]</code>	<p>Displays basic information about the contents of the <code>secmod.db</code> file.</p> <p>To display detailed information about a particular module including its slots and tokens, specify a value for <code>moduleName</code>.</p>
<code>-undefault <i>moduleName</i></code>	<p>Specifies the security mechanisms for which the named module will <i>not</i> be a default provider.</p> <p>You specify the security mechanisms by using the following option:</p> <p><code>-mechanisms <i>mechanismList</i></code></p>

Options

The following table describes what the options for `modutil` do.

Table 8-4 Options for `modutil`

Option	What the Option Does
<code>-ciphers <i>cipherList</i></code>	<p>Enables specific ciphers in a module that you are adding to the database.</p> <p><i>CipherList</i> is a colon-delimited list of cipher names. Enclose this list in quotation marks if it contains spaces.</p> <p>The following cipher is currently available:</p> <ul style="list-style-type: none"> • FORTEZZA
<code>-dbdir <i>dbFolder</i></code>	<p>Specifies a folder in which to access or create security module database files.</p> <p>On UNIX systems, the Security Module Database Tool defaults to the user's iPlanet or Netscape folder. Windows NT has no default folder, so you must use <code>-dbdir</code> to specify a folder.</p>
<code>-installdir <i>InstallationDir</i></code>	<p>Specifies the root installation folder for the files supplied via the <code>-jar JAR-file</code> command.</p> <p>The <i>InstallationDir</i> folder should be one in which it is appropriate to store dynamic library files—for example, a server root.</p>
<code>-libfile <i>libraryFile</i></code>	<p>Specifies a DLL (Dynamic Link Library) or library file containing the implementation of the PKCS #11 module that is being added to the database. Use a complete path to identify the file.</p>

Table 8-4 Options for modutil (*Continued*)

Option	What the Option Does
<code>-mechanisms <i>mechanismList</i></code>	<p data-bbox="686 284 1179 340">Specifies the security mechanisms for which a particular module will be the default provider.</p> <p data-bbox="686 357 1215 444">The <code>MECHANISM_LIST</code> is a colon-separated list of mechanism names. Enclose this list in quotation marks if it contains spaces.</p> <p data-bbox="686 461 1215 604">The module becomes a default provider for the listed mechanisms when those mechanisms are enabled. If more than one module is assigned as a mechanism's default provider, the mechanism's default provider is listed as undefined.</p> <p data-bbox="686 621 1215 651">The following mechanisms are currently available:</p> <ul data-bbox="686 673 1179 1133" style="list-style-type: none"> <li data-bbox="686 673 768 696">• RSA <li data-bbox="686 718 768 741">• DSA <li data-bbox="686 763 876 786">• RC2, RC4, RC5 <li data-bbox="686 808 768 831">• DES <li data-bbox="686 854 753 876">• DH <li data-bbox="686 899 839 921">• FORTEZZA <li data-bbox="686 944 782 966">• SHA1 <li data-bbox="686 989 822 1012">• MD2, MD5 <li data-bbox="686 1034 1162 1057">• RANDOM (for random number generation) <li data-bbox="686 1079 1179 1133">• FRIENDLY (for certificates that are publicly readable).
<code>-newpwfile <i>newPasswordFile</i></code>	<p data-bbox="686 1156 1208 1242">Specifies a text file containing a token's new password. This allows automatic updating of the password when using the <code>-changePW</code> command.</p>

Table 8-4 Options for modutil (*Continued*)

Option	What the Option Does
<code>-nocertdb</code>	Instructs <code>modutil</code> to not open the certificate or key databases. This has several effects: <ul style="list-style-type: none"> • When used with the <code>-changepw</code> command, no one will be able to set or change the password on the iPlanet internal module, because the password is stored in <code>key3.db</code>. • When used with the <code>-create</code> command, only a <code>secmod.db</code> file will be created; <code>cert7.db</code> and <code>key3.db</code> will not be created. • When used with the <code>-jar</code> command, signatures on the JAR file will not be checked.
<code>-pwfile passwordFile</code>	Specifies a text file containing a token's current password. This allows automatic entry of the password when using the <code>-changepw</code> command.
<code>-slot slotName</code>	Specifies a particular slot to enable or disable when using the <code>-enable</code> or <code>-disable</code> commands.
<code>-tempdir temporaryFolder</code>	Specifies a folder in which to store temporary files created by the <code>-jar</code> command. If a temporary folder is not specified, the current folder is used.

Usage

Tasks that you can perform using the `modutil` tool are listed here in the order you might do them, with the command and any options you use to perform them. 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 [-dbdir dbFolder]
```
- Displaying basic module information or detailed information about the contents of a given module:

```
-list [moduleName]
```

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

```
-add moduleName -libfile libraryFile [-ciphers cipherList]
[-mechanisms mechanismList]
```
- Adding a PKCS #11 module from a JAR file:

```
-jar JARfile -installdir installationFolder [-tempdir
temporaryFolder]
```
- Deleting a specific PKCS #11 module from a security module database:

```
-delete moduleName
```
- Initializing or changing a token's password:

```
-changepw token [-pwfile passwordFile][-newpwfile
newPasswordFile]
```
- Setting the default provider status of various security mechanisms in an existing PKCS #11 module:

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

```
-undefault moduleName -mechanisms mechanismList
```
- 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_or_false
```
- Disabling interactive prompts for the modutil tool in order to support scripted operation:

```
-force
```

JAR Information File

JAR (Java Archive) is a platform-independent file format that aggregates many files into one. JAR files are used by the `modutil` tool to install PKCS #11 modules. When `modutil` uses a JAR file, a special JAR information file must be included. This information file contains special scripting instructions and must be specified in the JAR file's `MANIFEST` file. Although the information file can have any name, you specify it by using the `Pkcs11_install_script METAINFO` command. To declare this `METAINFO` command in the `MANIFEST` file, include it in a text file that is passed to the iPlanet Signing Tool.

Sample METAINFO Tag and JAR Information File

If a PKCS #11 installer script was stored in the information file `pk11install`, the text file for the iPlanet Signing Tool would contain the following `METAINFO` tag:

```
+ Pkcs11_install_script: pk11install
```

The following is an example of a JAR information file which contains instructions for installing a PKCS #11 module on different platforms. The syntax used in the file is explained in “JAR Information File Syntax,” which begins on page 157.

```

ForwardCompatible { IRIX:6.2:mips SUNOS:5.5.1:sparc }
Platforms {
  WINNT::x86 {
    ModuleName { "Fortezza Module" }
    ModuleFile { win32/fort32.dll }
    DefaultMechanismFlags{0x00000001}
    CipherEnableFlags{0x00000001}
    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{0x00000001}
    CipherEnableFlags{0x00000001}
    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 }
  }
}

```

JAR Information File Syntax

Creating a JAR information file involves writing a script that specifies which tasks to perform when installing a module. In order to specify different module installation procedures for different platforms, you use *keys*, predefined commands and options that `modutil` interprets.

Keys are case-insensitive strings that are grouped into three categories:

- Global Keys
- Per-Platform Keys
- Per-File Keys

The following sections describe the function of each of these three categories and list the keys contained in each one.

Global Keys

Global keys define the platform-specific sections of the JAR information file. There are two global keys: `ForwardCompatible` and `Platforms`.

`ForwardCompatible` is an optional key that specifies a list of system architectures and operating systems that are compatible with later versions of the same architectures and operating systems. If the platform that `modutil` is installing the module on is not specified by the `Platforms` key, 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.

The `ForwardCompatible` key uses the following format:

```
ForwardCompatible { IRIX:6.2:mips SUNOS:5.5.1:sparc }
```

The platforms listed between the braces must have entries within the `Platforms` key.

`Platforms` is a required key that specifies 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. For more information, see “Per-Platform Keys” on page 159.

The platform string is in the following format:

```
system name:OS release:architecture.
```

On non-UNIX operating systems, `OS release` is an empty string.

The `modutil` program obtains the system name, OS release, and architecture values from the system on which the `modutil` tool is running using low-level code written by Netscape. The following system names and platforms are currently recognized by the low-level Netscape code:

- 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 an entry in the `Platforms` list.

`ModuleName` is a required key that specifies the common name for the module. This name acts as a reference to the module for Netscape Communicator, the `modutil` tool, servers, or any other program that uses the iPlanet security module database.

`ModuleFile` is a required key that names the PKCS #11 module file (DLL or .so) for this platform. The file name should be a path that is relative to the JAR file location.

`DefaultMechanismFlags` is an optional key that 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 string constants listed in Table 8-5. If you omit the `DefaultMechanismFlags` entry, the value defaults to 0x0.

Table 8-5 Mechanisms That You Can Specify Using `DefaultMechanismFlags`

Mechanism	Hexadecimal Bitstring Value
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` is an optional key that specifies ciphers that are provided by this module but not by iPlanet products. You use this key if you want to enable these ciphers for iPlanet products. The key is a bitstring specified in hexadecimal (0x) format. It is constructed as a bitwise OR of the following string constants. If you omit the `CipherEnableFlags` entry, the value defaults to 0x0. The only key that is provided right now is for Fortezza:

```
FORTEZZA:                0x00000001
```

`Files` is a required key that lists the files that need to be installed for this module. Each entry in the file list is a key-value pair. The key includes the path to the file that is contained in the JAR archive and the value list contains the attributes of the file. At a minimum, you must specify either `RelativePath` or `AbsolutePath` for each file. If desired, you can specify additional attributes. For more information, see “Per-File Keys” on page 160.

The `EquivalentPlatform` key specifies that the attributes of the named platform should also be used for the current platform. Using this key saves time when more than one platform uses the same settings.

Per-File Keys

These keys have meaning only within an entry in a `Files` list. At a minimum, `RelativePath` or `AbsolutePath` must be specified. If both are specified, the relative path is tried first, and the absolute path is used only if a relative root folder is not provided by `modutil`.

The `RelativePath` key specifies the destination path of the file, relative to a folder indicated at installation. You can assign values for two variables in the relative path, “%root%” and “%temp%”. At run time, “%root%” is replaced with a folder in which files should be installed, such as the server’s root folder. The “%temp%” folder 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 of these files could be installed in the temporary folder. Files destined for the temporary folder are in place before any executable file is launched. They are not deleted until all executable files have finished.

The `AbsolutePath` key specifies the destination of the file as an absolute path. If both `RelativePath` and `AbsolutePath` are specified, `modutil` attempts to use the relative path. If it is unable to determine a relative path, it uses the absolute path.

The `Executable` key specifies that a file is to be executed during the course of the installation. Typically this key is used to identify a setup program provided by a module vendor. The setup program itself is specified by the `RelativePath` or `AbsolutePath` key.

For example, to specify that the `setup.exe` program (located in the `%temp%` folder) is an executable file, you would include the following lines in your JAR information file:

```
Executable
RelativePath { %temp%/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 listed in the script file. Use the `Executable` key before a `RelativePath` or `AbsolutePath` key to indicate the order in which the files are to be run.

The `FilePermissions` key specifies the access permissions to apply to a file. The `modutil` program interprets the key as a string of octal digits, following the standard UNIX format. This key is a bitwise OR of the string constants listed in Table 8-6. For example, to specify Read and Execute access for all users, you would enter 555 (bitwise $400 + 100 + 040 + 010 + 004 + 001$).

The following table lists the file permissions that you can specify using `FilePermissions`.

Table 8-6 File Permissions That You Can Specify Using `FilePermissions`

File Permission	Bitstring Value
User Read	400
User Write	200
User Execute	100
Group Read	040
Group Write	020
Group Execute	010
Other Read	004
Other Write	002
Other Execute	001

Some platforms may not understand these permissions. The permissions are applied only if they make sense for the current platform. If this key is omitted, a default value of 777 (Read, Write and Execute for all users) is assumed.

Examples of Using modutil

This section includes examples of using `modutil` to perform the following tasks:

- 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

You could enter something like the following example to create a set of security management database files in a directory:

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

Before running this program, the `modutil` tool displays a warning:

```
WARNING: Performing this operation while an iPlanet product is
running could cause corruption of your security databases. If an
iPlanet 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 creates the databases and 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 uses `modutil` to retrieve detailed information about a specific module:

```
modutil -list "iPlanet Internal PKCS #11 Module" -dbdir C:\databases
```

The `modutil` tool displays information similar to this:

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

```
-----
Name: Netscape Internal PKCS #11 ModuleLibrary 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
...
```

Setting a Default Provider

You could enter something like the following example to make a specific module the default provider for the RSA, DSA, and RC2 security mechanisms:

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

Before running this program, the `modutil` tool displays a warning:

```
WARNING: Performing this operation while an iPlanet product is
running could cause corruption of your security databases. If an
iPlanet 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 makes the change and displays the following:

```
Using database directory C:\databases...
Successfully changed defaults.
```

Enabling a Slot

You could enter something like the following example to enable a particular slot in a module:

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

Before running this program, the modutil tool displays a warning:

```
WARNING: Performing this operation while an iPlanet product is
running could cause corruption of your security databases. If an
iPlanet 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 enables the slot and displays the following:

```
Using database directory C:\databases...
Slot "Cryptographic Reader" enabled.
```

Enabling FIPS Compliance

You could enter something like the following example to enable FIPS-140-1 compliance in iPlanet Administration Server's internal module:

```
modutil -fips true
```

Before running this program, the modutil tool displays a warning:

```
WARNING: Performing this operation while an iPlanet product is
running could cause corruption of your security databases. If an
iPlanet 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 enables FIPS compliance and displays the following:

```
FIPS mode enabled.
```

Adding a Cryptographic Module

You could enter something like the following example to add 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
```

Before running this program, the modutil tool displays a warning:

```
WARNING: Performing this operation while an iPlanet product is
running could cause corruption of your security databases. If an
iPlanet 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 adds the module and displays the following:

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

Installing a Cryptographic Module From a JAR File

You could enter something like the following example to install a cryptographic module from an installation script. The example uses this script:

```
Platforms {
  WinNT::x86 {
    ModuleName { "SuperCrypto 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
-installldir "C:\winnt"
```

Before running this program, the modutil tool displays a warning:

WARNING: Performing this operation while an iPlanet product is running could cause corruption of your security databases. If an iPlanet 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 installs the module and 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=SuperCrypto 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)

After you press y, the tool displays the following:

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 ./pkllinst.dir/setup.exe

Executing "./pkllinst.dir/setup.exe"...

"./pkllinst.dir/setup.exe" executed successfully

Installed module "SuperCrypto Module" into module database

Installation completed successfully

Changing the Password on a Token

You could enter something like the following example to change the password for a security device in use by a module.

```
C:\modutil> modutil -dbdir "C:\databases" -changepw "Administration  
Server Certificate DB"
```

Before running this program, the modutil tool displays a warning:

```
WARNING: Performing this operation while an iPlanet product is  
running could cause corruption of your security databases. If an  
iPlanet 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 changes the password and displays the following:

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

```
Enter old password:
```

After you enter the old password, the tool displays the following:

```
Enter new password:
```

After you enter the new password, the tool displays the following:

```
Re-enter new password:
```

After you re-enter the new password, the tool displays the following:

```
Token "Administration Server Certificate DB" password changed  
successfully.
```

modutil

Advanced Server Management

Chapter 9, “Access Control”

Chapter 10, “Using SSL and TLS with iPlanet Servers”

Chapter 11, “Using SNMP to Monitor Servers”

Access Control

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

- Overview of Access Control
- Working With Access Control Instructions

Overview of Access Control

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

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

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

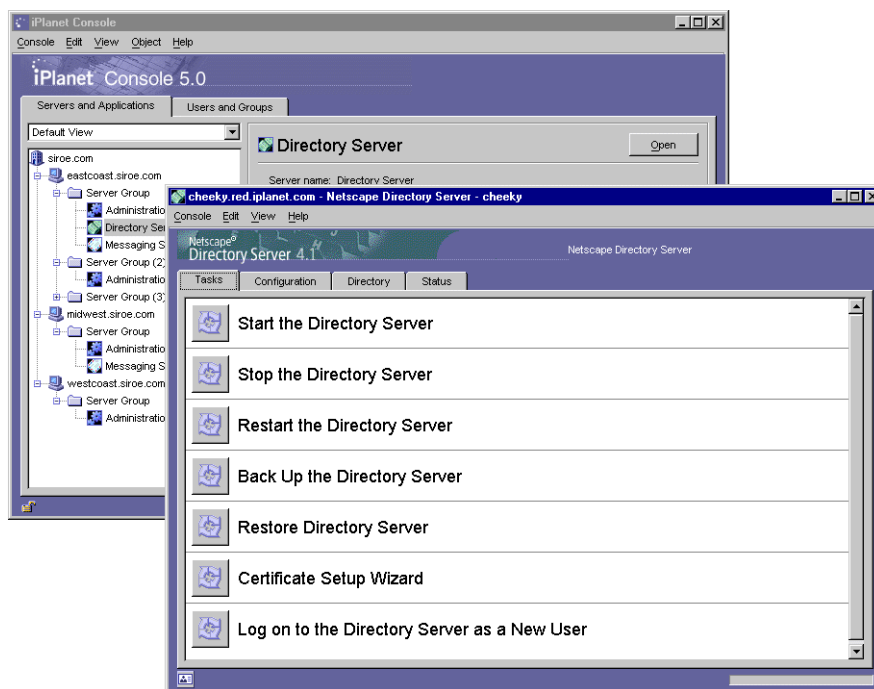
For detailed information about ACIs for a particular iPlanet or Netscape server, see the documentation for that server.

Examples of Access Control

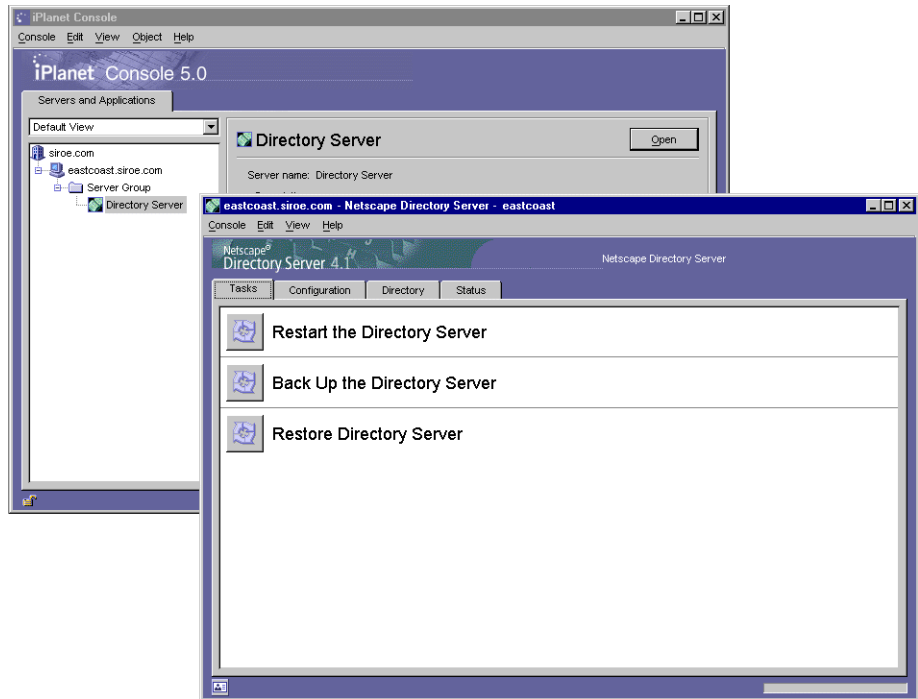
The following examples illustrate how an organization might use ACIs to grant and restrict access to servers and data by different administrators.

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

Figure 9-1 Jane's Unrestricted View of Resources and Tasks



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

Figure 9-2 John's Restricted View of Resources and Tasks

Setting Access Permissions For Servers

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

To Set Access Permissions for a Server in the Navigation Tree

1. Select a server in the iPlanet Console navigation tree.
2. From the Object menu, choose Set Access Permissions.

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

3. In the Set Permissions dialog box, specify who has administrative access to the server.

To add a user to the list of people who can administer the server, click the Add User button, and then search for the user or group that you want to grant administrative rights to. For more information on locating users and groups in the directory, see “Locating a User or Group in the Directory” on page 89.

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

Note that granting a user the right to administer a server does not automatically allow that user to give others the same right. If you want to allow a user to grant administrative rights to other users, you must add him or her to the Configuration Administrators group. For instructions on how to do this, see “To Add Users to the Configuration Administrators Group” on page 101.

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

Working With Access Control Instructions

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

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

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

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

Each iPlanet or Netscape server may have its own uses for the ACI Editor and may have unique ACI extensions. For detailed information about a particular server’s ACI options, see the documentation for that server.

What's in an ACI

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

Target

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

Permissions

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

Bind Rules

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

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

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

```

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

```

Using the ACI Manager and ACI Editor

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

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

To Specify What You Want an ACI to Apply To

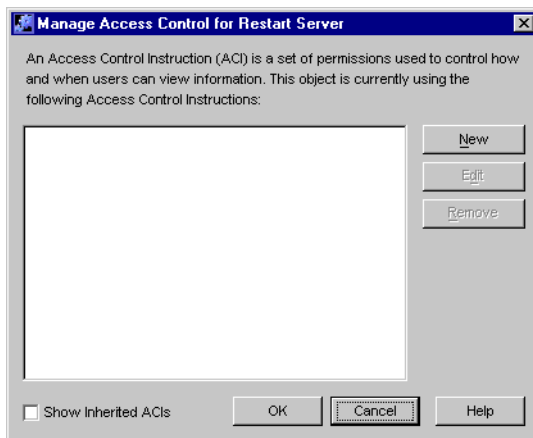
1. Select an object that you want to apply ACIs to.
 - To select a task or directory entry click its name.

Select a task name in an individual server management window. Select a directory entry in the Directory tab of the iPlanet Directory Server management window.
 - To select a user interface (UI) element, choose Preferences from the Edit menu, and then click the UI Permissions tab. On the tab, select an onscreen element from the list.
2. Open the ACI Manager.
 - To open the ACI Manager from a server management window, right-click and choose Set Access Permissions.

- To open the ACI Manager from the UI Permissions panel of the Preferences dialog box, click the Permissions button.
- In some servers, you can also open the ACI Manager by choosing Set Access Permissions from the Edit or Object menu.

The default ACI Manager window is shown in Figure 9-3:

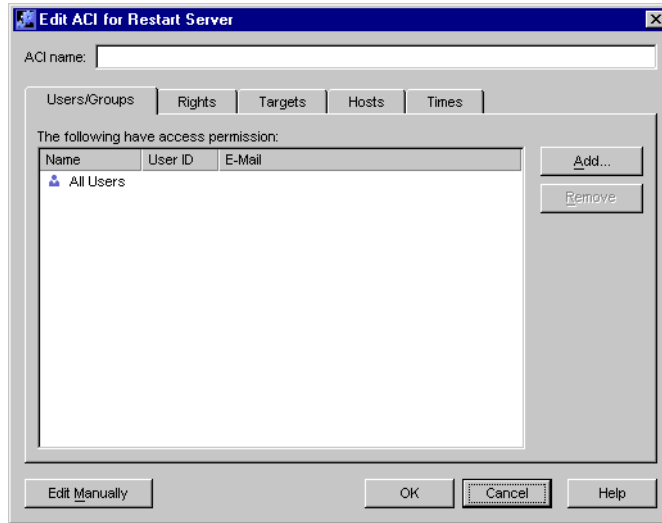
Figure 9-3 Default ACI Manger Window



To Create a New ACI With the Visual ACI Editor

1. In the ACI Manager click New.

The ACI Editor appears.



2. Enter a name for this ACI in the ACI Name field.
3. On the Users/Groups tab, click Add.
4. Identify the users, groups, or administrators to which you want to grant access.
 - o First, search for users, groups, or administrators to grant access to:

Search. From this drop-down list, select a set of entries in which you want to search. You can choose Administrators, Users, Groups, or “Users and Groups.”

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

Search. Click this button to perform your search.

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

- Then, grant access:

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

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

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

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

5. Click OK.

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

Check All. Click to select all rights.

Check None. Click to deselect all rights.

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

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

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

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

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

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

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

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

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

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

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

8. On the Hosts tab, click Add.
9. Enter the host name or IP address that you want to grant access to, then click OK. You can use the * wildcard when specifying hosts.
10. On the Times tab, select the times during which you want to grant access to the desired users, groups, and hosts.

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

11. Click OK to save this ACI.

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

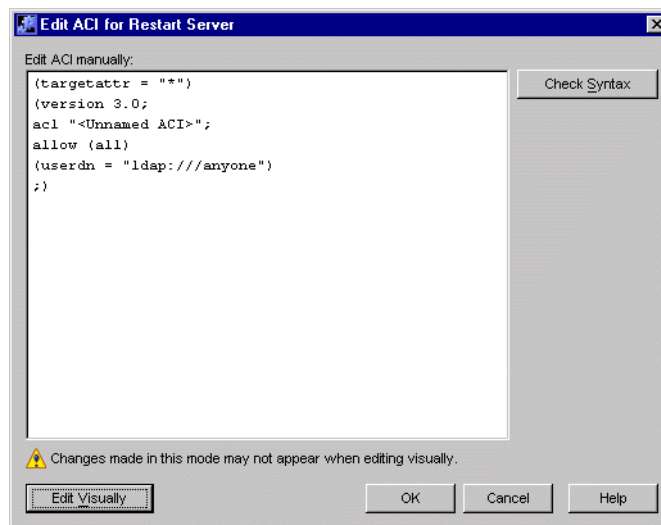
To Create a New ACI With the Manual ACI Editor

1. In the ACI Manager click New.

The ACI Editor appears.

2. Enter a name for this ACI in the ACI Name field.
3. Click Edit Manually.

The ACI Editor switches into manual mode.



4. Enter your ACI.

For more information on creating ACIs, see the *Directory Server Administrator's Guide*.

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

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

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

If you selected a task or directory entry (in “To Specify What You Want an ACI to Apply To” on page 176), the ACI is automatically applied to it. If you selected a user interface element, you must restart iPlanet Console for the ACI to take effect.

To Edit an Existing ACI With the ACI Editor

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

The ACI Editor appears.

2. Make the desired changes.

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

3. When you are finished, click OK.

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

To Remove an ACI

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

2. Click Remove.

3. Click OK to remove the ACI.

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

Using SSL and TLS with iPlanet Servers

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

This chapter contains the following sections:

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

The SSL and TLS Protocols

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

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

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

SSL and TLS Ciphers

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

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

Choosing SSL and TLS Ciphers

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

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

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

For detailed information on determining which cipher suites to use when setting up SSL, see Appendix C, “Introduction to SSL,” which begins on page 269.

Preparing to Use SSL and TLS Encryption

All iPlanet servers, as well as Netscape 4.x servers, support PKCS #11 and the SSL protocol. Many iPlanet servers also support TLS. Before you request certificates and begin to exchange information securely, you’ll need to set up SSL and TLS. If you’re using an external security device, you will also need to install a PKCS #11 module.

Using External Security Devices

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

A PKCS #11 module is a device, implemented in hardware or software, that provides cryptographic services such as encryption, decryption and, in some cases, storage of keys and certificates. All iPlanet and Netscape servers include a built-in software PKCS #11 module. Other kinds of PKCS #11 modules include the FORTEZZA module, used by the United States government, and the Litronic cryptographic module for smart card readers.

Netscape and iPlanet servers can use a variety of external PKCS #11 modules provided by different manufacturers. Before using an external module, you must install the manufacturer’s drivers on the machine running your Netscape or iPlanet server.

Slots and Security Devices

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

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

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

To Install an External Security Device

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

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

2. In the iPlanet Console navigation tree, select the server instance that you want to use the PKCS #11 module with, and then click Open.
3. From the server's Console menu, choose Security > Configure Security Modules, and then click Install.
4. In the Install Security Module dialog box, enter the following information:
 - Enter the PKCS #11 module driver filename.** Enter the full path to the driver file that came with your device. This file will have the extension DLL, JAR, SO, or sl.
 - Enter an identifying name for this module.** Enter a descriptive name that will help you identify this device.
5. Click OK, and then click Close.

To Remove an External PKCS #11 Module

1. In the iPlanet Console navigation tree, select the server instance that is using the external PKCS #11 device, and then click Open.
2. From the server's Console menu, choose Security > Configure Security Modules.
3. Select your device from the list and then click Remove.
4. Click OK to confirm that you want to remove the device, and then click Close.

Obtaining and Installing a Server Certificate

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

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

SSL Certificates

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

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

For information on installing a server certificate, see “Generating a Server Certificate Request”, on page 188 and “Installing the Certificate”, on page 191.

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

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

To obtain a trusted CA certificate, first go to the internal certificate server or CA's web site. Copy the necessary certificate information and save it to a file. Then use the Certificate Installation Wizard to install the certificate. For more information, see “Installing the Certificate”, on page 191.

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

Preparing to Set Up SSL and TLS

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

Setting up SSL or TLS With an Internal Security Device

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

Setting up SSL or TLS With an External Security Device

To set up SSL with an external security device, such as FORTEZZA, first install the PKCS #11 module provided by the external device manufacturer. Then run the Certificate Request Wizard, specifying the external security device when prompted. For more information, see “To Install an External Security Device”, on page 186.

Setting Up SSL With Internal and External Security Devices

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

Generating a Server Certificate Request

You can use iPlanet Console to generate a certificate request which you can then submit to a CA.

To Generate a Certificate Request

1. In the iPlanet Console navigation tree, select the server instance with which you want to use SSL encryption.
2. Double click the server instance or click Open to open the management window for the server instance.

3. From the Console menu, choose Security > Manage Certificates.
Alternatively, you can click the Manage Certificates task.
4. Click Request to open the Certificate Request Wizard.
5. Choose “Request Certificate Manually,” and then click Next.
6. Enter the requested information:
 - Server Name.** (Optional) Enter the fully qualified hostname of the machine for which you’re requesting a certificate.
 - Organization.** (Optional) Enter your organization’s name.
 - Organizational Unit.** (Optional) Enter your division, department, or other organizational unit.
 - City/locality.** (Optional) Enter the city or locality in which your organizational unit is located.
 - State/province.** (Optional) Enter the state or province in which your organizational unit is located.
 - Country/region.** (Optional) Select the state or province in which your organizational unit is located, from the drop down menu.

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

 - Show DN.** Click to show the requestor information in distinguished name (DN) format. The Show DN button is visible only when you are entering information in fields.
 - Show Fields.** Click to show the requestor information in fields. The Show Fields button is visible only when you are entering information in DN format.
7. Click Next.
8. Enter the password for the security device that will store this certificate.

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

NOTE Important: You need to generate a different request for each device.

9. Click Next.
10. Select one of the following:
 - Copy to Clipboard.** Click to copy your certificate request to the clipboard.
 - Save to File.** Click to save your request as a text file. You will be prompted to choose a name and location for the file.

The certificate request you have copied to the clipboard or saved as a text file is required to email to a Certificate Authority issuing the new certificate. See “To Send a Server Certificate Request as Email”, on page 190.
11. Click Done to close the Certificate Request Wizard.

Sending a Server Certificate Request

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

To Send a Server Certificate Request as Email

1. Use your email program to create a new email message.
2. Paste your certificate request into the message.
 - If you copied the certificate request to the clipboard, paste it into the body of the message.
 - If you saved your certificate request to a file, open it in a text editor. Copy and paste the request into the body of the message.
3. Enter a subject and recipient for your request. The type of subject and recipient varies depending on which CA you are using. For more information, see your CA’s web site.
4. Send the email message to the CA.

Once you’ve submitted your request, you must wait for the CA to respond with your certificate. Turnaround time is highly variable and depends on the CA. If your company has an internal CA, it may take only a day or two to receive your certificate. If you are using an external CA, it could take as long as several weeks for that CA to respond to your request.

Installing the Certificate

Depending on the CA, you may receive your certificate in an email message or you may have to retrieve it from the CA's web site. Once you have the certificate, you can back it up and install it.

To Back Up a Certificate

- Save, in a text file, the certificate data you received from the CA.

If you ever lose the certificate data, you can reinstall the certificate using this backup file.

To Install a Server Certificate

1. In the iPlanet Console navigation tree, select the server instance on which you want to install the certificate.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.

Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.

4. Click the Server Certs tab.
5. Specify where to store this certificate.
 - If you want to store this certificate on the internal security device, select internal (software) from the Security Device drop-down list, and then click Install.
 - If you want to store this certificate on an external hardware device, select the device from the Security Device drop-down list, and then click Install.
6. Enter the certificate's location or enter its text.

In this local file. If your certificate is stored in a text file on your system, enter the full path to the file.

In the following encoded text block. If you copied your certificate to the clipboard, paste the certificate's text into the text field by clicking the Paste from Clipboard button.

7. Click Next.

If the certificate information you entered above is valid, you see a page containing the details of your certificate.

8. Verify that the certificate information is correct, and then click Next.
9. Enter a name for the certificate, and then click Next.
10. Enter the password for the security device that will hold this certificate.

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

11. Click Done.

To Install a CA Certificate or Server Certificate Chain

1. Obtain the CA certificate or Server Certificate Chain from your CA.
2. In the iPlanet Console navigation tree, select the server instance on which you want to install the CA certificate.
3. Click Open to open the management window for the server instance.
4. On the Tasks tab, click the Manage Certificates task button.

Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.

5. Select the CA Certs tab, and then click Install.
6. Enter the certificate's location or enter its text:

In this local file. If the certificate is stored in a text file on your system, enter the full path to the file.

In the following encoded text block. If you copied the certificate to the clipboard, paste the certificate's text into the text field by clicking the Paste from Clipboard button.

7. Click Next.

If the certificate information you entered in step 6 is valid, you see a page containing the details of the certificate.

8. Verify that the certificate information is correct, and then click Next.
9. Enter a name for the certificate, and then click Next.

10. Select the trust options for this certificate:

Accepting Connections from Clients. Check this box if you want to trust client certificates issued by this CA.

Making Connections to Other Servers. Check this box if you want to trust server certificates issued by this CA.

11. Click Done.

Backing Up and Restoring Your Certificate Database

Whenever you install a certificate, you should back up your certificate database. If your database ever becomes corrupted, you can restore your certificate information from this backup.

To Back Up Your Certificate Database

1. Open your server root folder.
2. Copy all files in the `alias` folder to another location (preferably on a different disk).

This folder includes your certificates as well as the private key for your trust database.

To Restore Your Certificate Database From a Backup

- Copy your backup files to the `alias` subfolder of your server root folder.

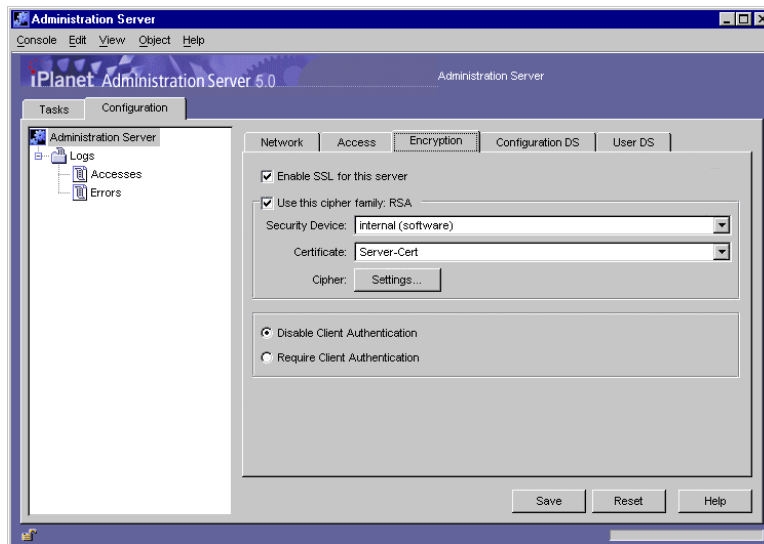
CAUTION If you restore your certificate database from a backup, any certificates that you installed after making the backup will be lost. Before restoring your certificate database, make sure that you have copies of all your certificates in case you need to reinstall them.

Activating SSL

Once you've obtained and installed a server certificate, use iPlanet Console to activate SSL on your iPlanet or Netscape 4.x server. The following procedure uses iPlanet Administration Server as its example. Activating SSL on other iPlanet and Netscape 4.x servers is done the same way, although in some cases the interface is slightly different. For more information on how to activate SSL on another server product, see that server's documentation.

To Activate SSL on an iPlanet Server or a Netscape 4.x Server

1. In the iPlanet Console navigation tree, select the server instance with which you want to use SSL encryption.
2. Click Open to open the management window for the server instance.
3. Click the Configuration tab.
4. Click the Encryption tab.



5. Enter information as appropriate:

Enable SSL for this server. Select this option if you want to secure this server with Secure Sockets Layer (SSL) encryption. All other SSL encryption options listed here become available to you only when you enable SSL by checking this box.

Use this cipher family. When you enable SSL encryption, the cipher families available to you are listed here. iPlanet Console currently supports two cipher families: RSA and Fortezza. The internal security device supports only RSA. If you're using a Fortezza card, you'll also see the Fortezza cipher family listed in the Encryption tab. Select the cipher families you want to use.

Security Device. Choose internal (software) if the key is stored in the local key database. All other choices on this list are available only if you are using an external module.

Certificate. Choose a server certificate to use with this server.

Settings. Click this button to modify cipher (encryption algorithm) settings for the certificate you selected above.

Disable Client Authentication. Select this option if you do not want this server instance to perform client authentication.

Require Client Authentication. Select this option if you want this server instance to require client authentication during the SSL handshake.

If you select this option, each iPlanet Console administrator will be prompted for a certificate when logging in. This ensures system security because all administrators must present acceptable certificates before they can perform management tasks. Even if an intruder obtains a user name and password, he or she will need to present a valid certificate (one issued by a trusted CA) to gain access to your enterprise.

For more information on setting trust options for CA certificates, see "To Change the CA Trust Options," which begins on page 198.

6. Click Save.
7. Exit iPlanet Console and restart the server you have SSL-enabled from the command line.

You can now start iPlanet Console again and log in to work with the server through iPlanet Console.

Managing Server Certificates

Periodically, you may need to update information for your installed SSL certificates. From iPlanet Console, you can renew a server certificate as well as view and edit settings for all certificates installed on a server.

Renewing a Certificate

Like credit cards or any other form of identification, all certificates have validity periods. You can check any certificate's expiration date from within iPlanet Console. When a server certificate is nearing its expiration date, you can use iPlanet Console to generate a renewal request.

To Check a Certificate Expiration Date

1. In the iPlanet Console navigation tree, select the server instance that is using the certificate whose expiration date you want to check.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.
Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.
4. Depending on which type of certificate you are checking, click the Server Certs or CA Certs tab.
5. Locate the certificate you are checking.
The certificate's validity period ends on the date shown in the Expiration Date column.

To Generate a Certificate Renewal Request

1. In the iPlanet Console navigation tree, select the server instance that is using the certificate you want to renew.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.
Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.
4. Click the Server Certs tab.

5. From the list of available certificates, select the one you want to renew, and then click the Renew button.
6. Select “Request Certificate Manually,” and then click Next.
7. Enter the requested information:

Server Name. (Optional) Enter the fully qualified hostname of the machine for which you’re requesting a certificate.

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

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

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

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

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

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

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

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

8. Click Next.
9. Enter the password for the security device that will store this certificate.

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

10. Click Next.
11. Copy or save the request in one of the following ways:
 - Copy to Clipboard.** Click to copy your certificate request to the clipboard.
 - Save to File.** Click to save your request as a text file. You will be prompted to choose a name and location for the file.

The certificate request you have copied to the clipboard or saved as a text file is required to email to a Certificate Authority issuing the new certificate. See “To Send a Server Certificate Request as Email” on page 190.
12. Click Done to close the Certificate Request Wizard.

You can now send your certificate renewal request to your CA. For more information, see “To Send a Server Certificate Request as Email” on page 190.

Changing the CA Trust Options

At times, you may need to reject a generally trusted CA. For example, if you are notified that a CA is experiencing technical difficulties that prevent certificate authentication, you can temporarily reject the CA’s certificate. When you are informed that the problem has been resolved, you can begin trusting the certificate again.

To Change the CA Trust Options

1. In the iPlanet Console navigation tree, select the server instance on which you want to change a CA trust option.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.

Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.
4. Click the CA Certs tab and then, from the list of available CA certificates, select the CA certificate for which you want to change the trust options.
5. Click the Edit Trust button.

6. Set the following CA trust options:
 - Accepting connections from clients (Client Authentication).** Uncheck this box if you want to reject client certificates issued by this CA.
 - Making connections to other servers (Server Authentication).** Uncheck this box if you want to reject server certificates issued by this CA.
7. Click OK.

Changing Security Device Passwords

You should periodically change the passwords for your security devices.

To Change a Security Device Password

1. In the iPlanet Console navigation tree, select the server instance that is using the security device for which you want to change the password.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.

Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.

4. Choose a security device from the drop-down list.
5. Click Password.
6. In the Change Security Device Password dialog box, enter password information:
 - Old password.** Enter the password currently used with this device.
 - New Password.** Enter a new password.
 - New Password (again).** Enter the password again to confirm it.
7. Click OK.

Managing Certificate Lists

Certificate revocation lists (CRLs) and compromised key lists (CKLs) allow CAs to specify certificates and keys that client or server users should no longer trust.

If data in a certificate changes, a CA can revoke the certificate and list it in a CRL—for example, when a user changes offices or leaves an organization before his or her certificate expires. If a key is tampered with or otherwise compromised, a CA can list it in a CKL.

CRLs and CKLs are produced and periodically updated by a CA.

To Obtain a CRL or CKL From a CA

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

Once you've saved the CRL or CKL file to a local directory, you can add its contents to the trust database. Once you do this, your server will no longer trust the certificates or keys that are specified in the CRL or CKL file.

To View, Add, or Delete a CRL or CKL

1. In the iPlanet Console navigation tree, select the server instance that you want to work with.
2. Click Open to open the management window for the server instance.
3. On the Tasks tab, click the Manage Certificates task button.

Alternatively, you can open the Console menu, and then choose Security > Manage Certificates.

4. Choose a security device.

If the server instance is using only the internal (software) security device, it is automatically chosen for you. If you are using an external (hardware) module, choose it from the drop-down list.

5. Select the Revoked Certs tab.

Every CRL and CKL for the chosen device is listed along with the date it was generated and the date it will next be updated.

6. View, add, or delete a CRL or CKL.
 - o To view the contents of a CRL or CKL, select its name, and click Detail.
 - o To add a CRL or CKL for the selected device, click Add, and then enter the following information:

Enter full path to CRL/CKL file. Provide the full path to the file containing the CRL or CKL.

File contains a Certificate Revocation List (CRL). Select this option if you're adding a CRL.

File contains a Compromised Key List (CKL). Select this option if you're adding a CKL.

- To delete a CRL or CKL from the selected device's trust database, select it, and then click Delete.
7. Click OK.

Using Client Authentication

You can configure some iPlanet and Netscape servers to require that clients present certificates when logging in. This allows a server to verify a client's authenticity and to determine if a user has access to the server. The process of presenting and verifying a client certificate is called client authentication.

This section tells you how to set up and use client authentication on your iPlanet or Netscape server. Before reading this section, check your server's documentation to verify that the server supports client authentication.

How Client Authentication Works

When a server receives a request from a client, it can ask for the client's certificate before proceeding. A Netscape client, such as Navigator or Communicator, is programmed to respond by sending a client certificate to the server.

After checking that a client certificate chain ends with a trusted CA, an iPlanet or Netscape server can optionally determine which user is identified by the client certificate and then look up that user's entry in the directory. The server authenticates the user by comparing the information in the certificate with the data in the user's directory entry.

In order to locate user entries in the directory, a server must know how to interpret certificates from different CAs. You provide the server with interpretation information by editing a file called `certmap.conf`. This file provides three kinds of information for each listed CA:

- It maps the distinguished name (DN) in the certificate to a branch point in the LDAP directory.
- It specifies which DN values from the certificate (user name, email address, and so on) the server should use for the purpose of searching the directory.

- It specifies whether the server should go through an additional verification process. This process involves comparing the certificate presented by the client for authentication with the certificate stored in the user's directory entry. By comparing the certificate, the server determines whether to allow access or whether to revoke a certificate by removing it from the user's directory entry.

If more than one directory entry contains the information in the user's certificate, the server can examine all matching entries in order to determine which user is trying to authenticate. When examining a directory entry, the server compares the presented certificate with the certificate stored in the entry. If the presented certificate doesn't match any directory entries or if matching entries don't contain matching certificates, client authentication fails.

After the server finds a matching entry and certificate in the directory, it can determine the appropriate kind of authorization for the client. For example, some servers use information from a user's entry to determine group membership, which in turn can be used during evaluation of ACIs to determine what resources the user is authorized to access.

You can also configure client authentication between an instance of Administration Server and another iPlanet or Netscape server. For more information see "Using Client Authentication Between Servers."

Preparing to Use Client Authentication

In order to accept certificates for client authentication, you must fulfill the following requirements:

- The server must have SSL turned on. For more information, see "Activating SSL" on page 194.
- The instance of Administration Server must trust the CA who issued the certificate to the client. For more information, see "Changing the CA Trust Options" on page 198.
- If you are going to search the directory for information contained in certificates, you must map specific CAs to branches of the user directory. To do this, you must edit a file called `certmap.conf`. The rest of this section describes this file and tells you how to edit it.

The certmap.conf File

When a server performs client authentication, it interprets a certificate, extracts user information, and then searches the directory for that information. In order to process certificates from different CAs, the server uses a file called `certmap.conf`. This file contains instructions on how to interpret different certificates and how to search the directory for the information that those certificates contain.

The `certmap.conf` file is stored in the `<server_root>/shared/config` folder. The file contains a default mapping as well as mappings for specific CAs.

The default mapping specifies what the server should do if a client certificate was issued by a CA that isn't listed in `certmap.conf`. The mappings for specific CAs specify what the server should do for client certificates issued by those CAs. All mappings define the following:

- Where in the directory the server should begin its search
- What certificate attributes the server should use as search criteria
- Whether the server should verify the certificate with a certificate that is stored in the directory

Mappings have the following syntax:

```
certmap name issuerDN
name:property [value]
name:property [value]
...
```

The first line of a mapping specifies the mapping's name as well as the DN for the issuer of the client certificate. You can name a mapping whatever you want, but the `issuerDN` must exactly match the issuer DN of the CA that issued the client certificate. For example, the following two `issuerDN` lines differ only in the number of spaces they contain, but the server would treat these two entries as different:

```
certmap moz ou=iPlanet CA,o=iPlanet,c=US
certmap moz ou=iPlanet CA, o=iPlanet, c=US
```

The second and subsequent lines of a mapping identify the rules that the server should use when searching the directory for information extracted from a certificate. These rules are specified through the use of one or more of the following **properties**: `DNComps`, `FilterComps`, `VerifyCert`, `CmapLdapAttr`, `Library`, and `InitFn`. These properties are explained next.

DNComps

`DNComps` is a comma-separated list of relative distinguished name (RDN) keywords used to determine where in the user directory the server should start searching for entries that match the information for the owner of the client certificate. The server gathers values for these keywords from the client certificate and uses the values to form a DN, which determines where the server starts its search in the directory.

For example, if you set `DNComps` to use the `o` and `c` RDN keywords, the server starts the search from the `o=org, c=country` entry in the directory, where `org` and `country` are replaced with values from the DN in the certificate.

- If there isn't a `DNComps` entry in the mapping, the server uses either the `CmapLdapAttr` setting or the entire subject DN in the client certificate to determine where to start searching.
- If the `DNComps` entry is present but has no value, the server searches the entire directory tree for entries matching the filter specified by `FilterComps`.

The following RDN keywords are supported for `DNComps`: `cn`, `ou`, `o`, `c`, `l`, `st`, `e`, and `mail`. You can list the keywords in lower case or upper case. You can use `e` or `mail`, but not both.

FilterComps

`FilterComps` is a comma-separated list of RDN keywords used to create a filter by gathering information from the user's DN in the client certificate. The server uses the values for these keywords to form the search criteria for matching entries in the LDAP directory. If the server finds one or more entries in the directory that match the user's information gathered from the certificate, the search is successful and the server performs a verification (if `verifycert` is set to `on`).

For example, if `FilterComps` is set to use the `e` and `uid` attribute keywords (`FilterComps=e,uid`), the server searches the directory for an entry whose values for `e` and `uid` match the user's information gathered from the client certificate. Email addresses and user IDs are good filters because they are usually unique entries in the directory.

The filter needs to be specific enough to match one and only one entry in the directory. The following RDN keywords are supported for `FilterComps`: `cn`, `ou`, `o`, `c`, `l`, `st`, `e`, and `mail`. You can list the keywords in lowercase or uppercase letters. You can use `e` or `mail`, but not both.

VerifyCert

`VerifyCert` tells the server whether it should compare the client's certificate with the certificate found in the user's directory entry. It takes one of two values: `on` or `off`. Setting the value to `on` ensures that the server will not authenticate the client unless the certificate presented exactly matches the certificate stored in the directory. Setting the value to `off` disables the verification process.

CmapLdapAttr

`CmapLdapAttr` is the name of the attribute in the directory that contains subject DN's 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 a `certmap.conf` mapping, the server searches the entire directory for an entry that contains the subject's full DN. The search criteria are the attribute named by `CmapLdapAttr` and the subject's full DN as listed in the certificate. If the search doesn't yield any entries, the server retries the search using the `DNComps` and `FilterComps` mappings. The search will take place more quickly if the attribute specified by `CmapLdapAttr` is indexed. For more information on indexing attributes, see the *Directory Server Administrator's Guide*.

Using `CmapLdapAttr` to match a certificate to a directory entry is useful when it's difficult to match entries using `DNComps` and `FilterComps`.

Library

`Library` is the pathname to a shared library or DLL. You need to use this property only if you want to extend or replace the standard functions that map information in `certmap.conf` to entries in the directory. This property is typically not necessary unless you have very specialized mapping requirements.

InitFn

`InitFn` is the name of an `init` function from a custom library. You need to use this property only if you want to extend or replace the functions that map information in `certmap.conf` to entries in the directory. This property is typically not necessary unless you have very specialized mapping requirements.

Custom Properties

You can use the Certificate Mapping API to create your own properties. For information on using the Certificate Mapping API, see “Certificate Mapping SDKs” at the following URL:

<http://developer.netscape.com/software/certificate/sdks.html>.

Editing the certmap.conf File

This section tells you how to edit the `certmap.conf` file.

To Edit the certmap.conf File

1. In a text editor, open `Server_Root/shared/config/certmap.conf`.
2. If necessary, make changes to the default mapping.

For example, you may want to change the value for `DNComps` or `FilterComps`. If you want to comment out a line, insert a `#` before it.

3. If desired, create a mapping for a specific CA.

The mapping should take this form: `certmap mappingName issuerDN`.

For example, to create a mapping named “Siroe CA” which has the issuer DN `ou=Siroe CA, o=Siroe, c=US`, you would enter the following:

```
certmap Siroe CA    ou=Siroe CA, o=Siroe, c=US
```

4. Add property settings for a specific CA’s mapping.

If you are using the `library` and `InitFn` properties, you must specify them before adding any additional properties.

When adding a property, use this form:

```
mappingName:propertyName value
```

For example, you could add a `DNComps` value of `o, c` for Siroe CA by entering the following line:

```
Siroe CA:DNComps    o, c
```

If you are using the `Library` and `InitFn` properties, a complete mapping might look like this:

```
certmap Siroe CA    ou=Siroe CA, o=Siroe, c=US
```

```

Siroe CA:Library      /usr/iplanet/server/userdb/plugin.so
Siroe CA:InitFn       plugin_init_dn
Siroe CA:DNComps     o, c
Siroe CA:FilterComps e, uid
Siroe CA:VerifyCert  on
Siroe CA:CmapLdapAttr certSubjectDN

```

5. Save the `certmap.conf` file.

Example certmap.conf Mappings

The following examples illustrate three different ways you can use the `certmap.conf` file.

Example of a Default Mapping

Here are the contents of a simple `certmap.conf` file that contains only the default mapping:

```

certmap default      default
default:DNComps     ou, o, c
default:FilterComps e, uid
default:verifycert  on

```

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

The server then uses the values for *e* (email address) and *uid* (user ID) from the certificate to search for a match in the directory before authenticating the user. When it finds a matching entry, the server verifies the certificate by comparing the certificate the client sent to the certificate stored in the directory.

Example of an Additional Mapping

Here are the contents of a sample `certmap.conf` file that defines a default mapping as well as a mapping for MyCA:

```

certmap default      default
default:DNComps
default:FilterComps e, uid

```



```
certmap MyCA          ou=MySpecialTrust,o=MyOrg,c=US
MyCA:DNComps         ou,o,c
MyCA:FilterComps     e
MyCA:verifycert      on
```

When the server gets a certificate from a CA other than MyCA, the server uses the default mapping, which starts at the top of the directory tree and searches for an entry matching the client's email address (e) and user ID (uid). If the certificate is from MyCA, the server starts its search at the directory branch containing the organizational unit specified in the subject DN and searches for email addresses (e) that match the one specified in the certificate. If the certificate is from MyCA, the server verifies the certificate. If the certificate is from another CA, the server does not verify it.

Example of a Mapping With an Attribute Search

This example uses the `CmapLdapAttr` property to search the directory for an attribute called `certSubjectDN` whose value exactly matches the entire subject DN in the client certificate:

```
certmap MyCo          ou=My Company Inc, o=MyCo, c=US
MyCo:CmapLdapAttr     certSubjectDN
MyCo:DNComps          o, c
MyCo:FilterComps      mail, uid
MyCo:verifycert       on
```

If the subject DN in the client certificate is `uid=Henry Jones Junior, o=Siroe Inc, c=US`, then the server searches for entries that have `certSubjectDN=uid=Henry Jones Junior, o=Siroe Inc, c=US`.

If one or more matching entries are found, the server proceeds to verify the entries. If no matching entries are found, the server uses `DNComps` and `FilterComps` to search for matching entries. For the client certificate described above, the server would search for `uid=Henry Jones Junior` in all entries under `o=Siroe Inc, c=US`.

Using Client Authentication Between Servers

If both servers support client authentication, you can use client authentication when establishing a connection from one iPlanet or Netscape server to another. Typically, you use client authentication to authenticate an instance of Administration Server to another iPlanet or Netscape server instance. In these cases, the instance of Administration Server acts as the client.

The following procedure tells you how to set up client authentication between an iPlanet or Netscape server and an instance of Administration Server.

To Set Up Client Authentication Between Servers

1. Install certificates on an instance of Administration Server and the iPlanet or Netscape server instance that will perform the authentication.

For more information, see “To Install a Server Certificate” on page 191.

2. If necessary, install CA certificates and specify that they should be trusted.

The instance of Administration Server needs to trust the CA that issued the certificate in use by the iPlanet or Netscape server instance. The iPlanet or Netscape server instance needs to trust the CA that issued the certificate in use by the instance of Administration Server.

For more information, see “To Install a CA Certificate or Server Certificate Chain” on page 192.

3. On the iPlanet or Netscape server instance that will perform the authentication, enable SSL and Client Authentication, and then restart the server.

Typically, you enable SSL and Client Authentication by changing the encryption settings on the server’s Configuration tab. For more information, see your server’s documentation.

4. In a text editor, open `serverRoot/admin-serv/config/adm.conf`.
5. Change the value for `ldapPort` to the secure port in use by the iPlanet or Netscape server instance.
6. Restart the instance of Administration Server.

For more information, see “Restarting Administration Server” on page 113.

The iPlanet or Netscape server instance now uses client authentication when communicating with the instance of Administration Server.

Client Authentication for Users

You can use client authentication to verify the identity and access permission of a user, typically an administrator, to an Administration Server instance. Before enabling client authentication for users, the server must have a CA certificate chain and server certificate installed and have SSL enabled.

Instructions for obtaining and installing server certificates and CA certificate chains are found in this chapter in the section entitled “Obtaining and Installing a Server Certificate”. Instructions for enabling SSL are also found in this chapter in the section entitled “Activating SSL”.

NOTE New and existing certificates are not recognized by Administration Server unless they are stored in the Netscape Navigator 4.7X certificate database format. For initial setup of client authentication, store certificates in the Netscape Navigator 4.7X browser.

To Set Up Client Authentication for Users

1. Install certificates on both the instance of Administration Server and the client that will participate in authentication.

For more information, see “To Install a Server Certificate” on page 191.

2. If necessary, install CA certificates and specify that they should be trusted.

The instance of Administration Server needs to trust the CA that issued the certificate in use by the client. The client needs to trust the CA that issued the certificate in use by the Administration Server.

For more information, see “To Install a CA Certificate or Server Certificate Chain” on page 192.

3. On the Administration Server instance that will perform the authentication, enable SSL and Client Authentication, and then restart the server.

Typically, this is done by changing the encryption settings on the server’s Configuration tab. For more information, see your server’s documentation.

4. Save client certificates in the Netscape Communicator certificate database.

New or existing certificates saved in the Netscape Communicator certificate database adopt the appropriate database format.

5. Copy the Netscape Communicator certificate database files, `cert7.db` and `key3.db`, that contain your certificates to your `.mcc` directory.

In WindowsNT, the `cert7.db` and `key3.db` files are located in
`C:\ProgramFiles\netscape\Users\<username>`

In Unix, the `cert7.db` and `key3.db` files are located in your home directory,
`/$HOME/.netscape`. `$HOME` is your root directory if you are running
Administration Server as root. `$HOME` is your user home directory if you are
running Administration Server as a user, for example, `/u/<username>` or
`/home/<username>`.

In Windows NT the `.mcc` directory is located in
`C:\WINNT\Profiles\<username>`

In Unix the `.mcc` directory is located in your home directory. For example, if
the Administration Server is running as root, then `.mcc` directory is located in
the root directory, `/.mcc`. If Administration Server is running as a user, then
`.mcc` is in your user directory, `/u/<username>/.mcc` or
`/u/home/<username>/.mcc`.

The next time you start Console, the Select Certificate window appears. Select a
certificate from the pull down menu to continue with an encrypted session in
Console.

Using SNMP to Monitor Servers

You can use the Simple Network Management Protocol (SNMP) to manage your iPlanet and Netscape servers. This chapter explains how SNMP works and tells you how to set it up on your network. The chapter contains the following sections:

- SNMP Basics
- Setting Up SNMP on UNIX Systems
- Using a Proxy SNMP Agent on UNIX Systems
- Reconfiguring a Native Agent on UNIX Systems
- Starting the Master Agent on UNIX Systems
- Enabling the Subagent on UNIX Systems
- Using the Windows NT SNMP Service

SNMP Basics

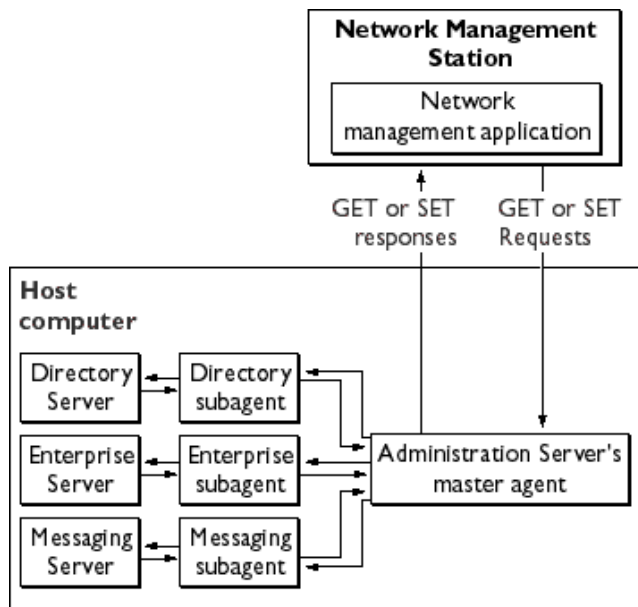
SNMP is a protocol used to exchange data about network activity. It defines a standard method of communication used to manage products from different vendors. This standard allows administrators to remotely manage hardware and software located across their network.

Each piece of controlled hardware and software is known as a managed device. A managed device is anything that runs SNMP, such as a host, router, or iPlanet server.

The machine used to monitor and configure managed devices is called a network management station. A network management station is usually a powerful workstation running network management applications which graphically show information about managed devices. For example, a network management application might show which servers in your enterprise are running and which are shut down, or the application might report the number and type of error messages received.

iPlanet and Netscape servers transmit data to a network management station using two types of agents: SNMP subagents and SNMP master agents. An SNMP subagent gathers information and sends it to an SNMP master agent. The SNMP master agent transfers the data to the network management station. Every iPlanet and Netscape server has an SNMP subagent except for iPlanet Administration Server, which either has a master agent (on UNIX) or no agent (on Windows NT).

A single machine can host multiple subagents, but a machine can only have one master agent. For example, if you have one instance each of Enterprise Server, Directory Server, and Messaging Server installed on one host, each will have its own subagent. All three subagents will report to the same master agent. This master agent is located on the same host machine as the subagents. Figure 11-1 illustrates this example.

Figure 11-1 Interaction Between a Network Management Station and a Host Computer

The Windows NT operating system includes an SNMP master agent. iPlanet Administration Server employs this service when utilizing SNMP. You can access and operate this master agent through the Network control panel. In the UNIX environment, the master agent is installed with Administration Server.

Some UNIX operating systems support an extended version of SNMP called the SNMP multiplexing protocol (usually known as SMUX). This allows iPlanet servers to operate without a master agent. For those versions of UNIX that do not support SMUX, you can use iPlanet Console to manage the master agent that iPlanet provides.

How SNMP Works

A managed device, such as a server, stores its configuration and management settings as variables. Some of these variables can be read and changed over SNMP while others cannot. The variables that the master agent can read and change are called managed objects. Managed objects are defined in a tree-like hierarchy known as a management information base (MIB).

Each iPlanet or Netscape server provides a management information base (MIB) for use in SNMP communication. This MIB contains managed objects pertaining to the server's operation. Each managed object has a unique object identifier. A server can report significant events to the network management station by sending "trap" messages (often called just "traps") containing these object identifiers. In addition, the network management station can initiate communication, and then specify one or more object identifiers when querying a server's MIB for data. The network management station can also remotely change variables in the MIB by specifying an object identifier and sending its new value.

iPlanet MIBs

Each iPlanet or Netscape server has its own MIB. All iPlanet MIBs are located in the `<server root>/plugins/snmp` directory.

A server's MIB contains variable definitions used when managing that particular server. Some of these variables can be modified over SNMP by a network management station while others are flagged as read-only or inaccessible. See your server's documentation for detailed information about its management variables.

The Administration Server MIB

iPlanet Administration Server stores its MIB in a file called `netscape-main.mib`.

The Administration Server MIB lists the object identifiers for all installed iPlanet servers. It also defines the object identifier shared by all iPlanet and Netscape servers. This object identifier is

```
netscape OBJECT IDENTIFIER ::= {enterprises 1450}
```

The `netscape-main.mib` file may look like this:


```

--
-- Netscape Main Mib for SNMP support
--

NETSCAPE-MIB DEFINITIONS ::=
BEGIN
    IMPORTS OBJECT-TYPE
            FROM SNMPv2-SMI
    MODULE-IDENTITY
            FROM SNMPv2-SMI
    enterprises
            FROM ObjectIds
    OBJECT-IDENTITY, Counter64
            FROM SNMPv2-SMI;

    netscape OBJECT IDENTIFIER ::= { enterprises 1450 }

-- All netscape sub-agents must branch off of the netscape root
-- above. Following objids for individual sub-agents have been
-- taken already.

-- http    OBJECT IDENTIFIER ::= { netscape 1 }
-- nsmail  OBJECT IDENTIFIER ::= { netscape 5 }
--

END

```

Types of SNMP Messages

SNMP defines three types of messages: GET, SET, and trap. The network management station uses GET messages to request data and SET messages to change variable values in the MIB. The messages sent by a server to the network management station are known as trap messages.

The following examples illustrate how a network management station, and the servers it communicates with, use GET, SET, and trap messages.

Network Management Station-Initiated Communication

A network management station can request information from a server or change the value of a variable stored in a server's MIB. For example:

1. The network management station sends a GET message to the Administration Server master agent. The GET message is a request for the number of Directory Server errors encountered since the server was last started.
2. The master agent forwards the message to the Directory Server's SNMP subagent.
3. The subagent retrieves the data.
4. The subagent sends the data to the master agent. The master agent sends a trap message containing the data to the network management station.
5. The network management station displays the data through its network management application.

Server-Initiated Communication

The server subagent sends a trap message to the network management station when a significant event has occurred. For example:

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

Setting Up SNMP on UNIX Systems

In general, to use SNMP on UNIX Systems you must have a master agent and at least one subagent installed and running on your system. You need to install a master agent before you can enable a subagent. Some UNIX systems have their own SNMP master agent. If your system has one of these *native agents*, you can either disable it or change the port number that it uses. If you disable the native agent, you will only be able to use the master agent included with Administration Server. If you change the port number that the native agent uses, you can use it alongside Administration Server's master agent.

The procedures for setting up SNMP are different depending upon your system. Table 11-1 provides an overview of the procedures to follow in various situations. The actual procedures are described in detail later in this chapter.

Before you begin, examine your system.

- Is your system already running an SNMP agent that's native to your operating system?
- If so, does your native SNMP agent support SMUX communication? If your native agent supports SMUX, you don't need to install a master agent. However, you do need to change the native agent's configuration.

If you are unsure of how to verify this information, see your system documentation.

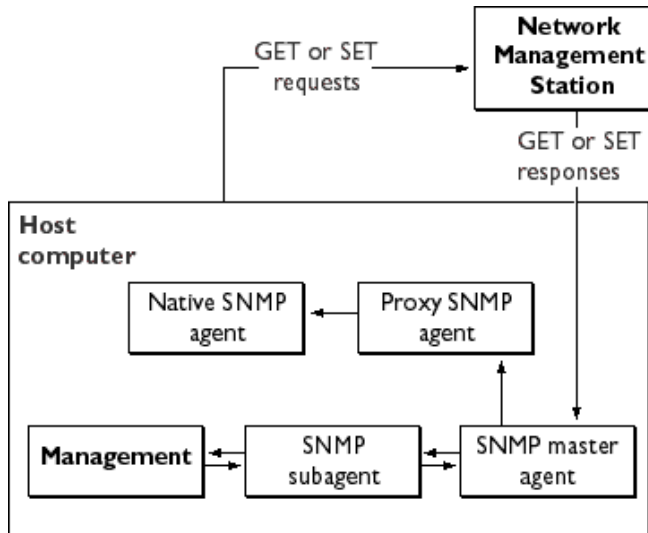
Table 11-1 Overview of Procedures for Enabling SNMP Master Agents and Subagents

If your server meets these conditions...	... follow these procedures
<ul style="list-style-type: none"> • The system does not have a native agent, or the native agent is not currently running. 	<ol style="list-style-type: none"> 1. Start the master agent. 2. Enable the subagent for each server installed on the system.
<ul style="list-style-type: none"> • The native agent is running, SMUX is not supported, and the system does not need to continue using the native agent. 	<ol style="list-style-type: none"> 1. Stop the native agent. 2. Start the master agent. 3. Enable the subagent for each server installed on the system.
<ul style="list-style-type: none"> • The native agent is running, SMUX is not supported, and the system needs to continue using the native agent. 	<ol style="list-style-type: none"> 1. Install and start a proxy SNMP agent. 2. Restart the native agent using a port number that is different from the master agent's port number. 3. Start the master agent. 4. Enable the subagent for each server installed on the system.
<ul style="list-style-type: none"> • The native agent is running and SMUX is supported. 	<ol style="list-style-type: none"> 1. Reconfigure the SNMP native agent. 2. Enable the subagent for each server installed on the system.

Using a Proxy SNMP Agent on UNIX Systems

If you want to use a native agent and the iPlanet Console master agent concurrently, you will need to set up a proxy agent. The proxy agent fields requests from the iPlanet master agent and then passes them on to the native agent. This scenario is illustrated in Figure 11-2.

Figure 11-2 Using a Proxy Agent When You're Running a Native SNMP Agent



In order to use both master agents simultaneously, you need to install and start the proxy SNMP agent. You also have to restart the native SNMP master agent using a port number other than the one used by the iPlanet Console master agent.

Installing and Starting the Proxy SNMP Agent

Before you install the proxy SNMP agent, make sure to stop the native master agent. See your system documentation for detailed instructions.

To Install the SNMP Proxy Agent

- Edit the `CONFIG` file located in the `<server-root>/plugins/snmp/sagt` directory so that it includes the port that the SNMP proxy agent will listen to. The file also needs to include the MIB trees and traps that the SNMP proxy agent will forward.

Here is a sample `CONFIG` file:

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES  1.3.6.1.2.1.1,
          1.3.6.1.2.1.2,
          1.3.6.1.2.1.3,
          1.3.6.1.2.1.4,
          1.3.6.1.2.1.5,
          1.3.6.1.2.1.6,
          1.3.6.1.2.1.7,
          1.3.6.1.2.1.8
FORWARD ALL TRAPS;
```

To Start the SNMP Proxy Agent

- At the command prompt, enter

```
sagt -c CONFIG&
```

After the proxy SNMP agent starts, you need to restart the native agent on the port you specified in the `CONFIG` file.

To Restart the Native Agent

- At the command prompt, enter

```
snmpd -P portNumber (specified in the CONFIG file)
```

For example, on the Solaris platform, using the port in the sample `CONFIG` file above, you would enter

```
snmpd -P 1161
```

Reconfiguring a Native Agent on UNIX Systems

If your native agent supports SMUX, you don't need to install a master agent. However, you do need to change the native agent's configuration.

UNIX uses several configuration files to screen its communications. One of them, `etc/snmp/conf/snmpd.conf`, needs to be changed so that the native agent accepts incoming messages from SMUX subagents. To change the file, add a line defining each subagent by its object identifier.

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

```
smux 1.3.6.1.4.1.1.1450.1 "" IPAddress netMask
```

where *IPAddress* is the IP address of the host on which the subagent is running and *netMask* is the network mask of that host (for instance, `255.255.0.0`).

NOTE Do not use the loopback address `127.0.0.1`; use the host's actual IP address instead.

For more information on configuring SNMP and SMUX, see the online manual page for `snmpd.conf`.

Configuring the Master Agent on UNIX Systems

In order to use SNMP, you must configure the master agent by specifying community strings and trap destinations.

Community Strings

A community string is a password that an SNMP agent uses for authorization.

A community string is a text string that an SNMP master agent uses for authorization. Whenever a network management station sends a message, it includes a community string. The agent receiving the message can then verify whether the network management station is authorized to obtain information. Community strings are not concealed when sent in SNMP packets; they are sent as ASCII text.

To ensure that a network management station is authorized to obtain information, the SNMP master agent compares the community string sent by the station to its list of accepted community strings. If the community string is listed, the network management station is authenticated.

Trap Destinations

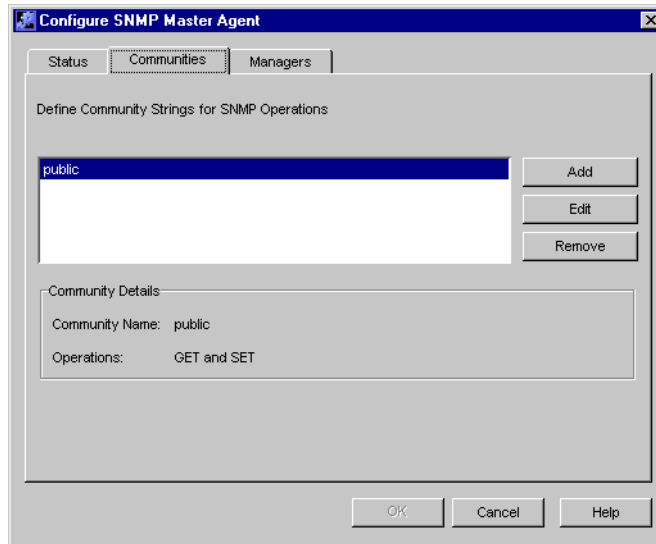
An SNMP trap is a message the SNMP agent sends to a network management station. For example, an SNMP agent might send a trap when a server goes down. The SNMP agent must know the address of the network management station in order to send traps. This address is called a trap destination.

Configuring the Master Agent using iPlanet Console

iPlanet Console provides an easy way to work with SNMP parameters. You can add, edit, and remove community strings and trap destinations from the Administration Server management window. You can also set the SNMP operations that a particular community string can request, as well as view any trap destinations you have already configured.

To Add, Edit, or Remove a Community String using iPlanet Console

1. In the iPlanet Console navigation tree, select the instance of Administration Server that you want to work with.
2. Click Open to open the management window for the server instance.
3. Click the Tasks tab.
4. Click the Configure SNMP Master Agent button, and then click Communities.



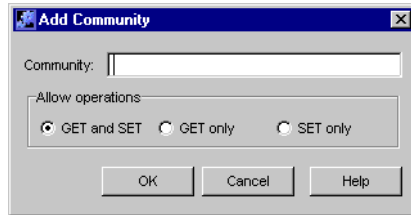
5. Click the appropriate button for the task you are performing.
 - If you want to add a community string, click Add.
 - If you want to edit a community string, select it, and then click Edit.
 - If you want to remove a community string, select it, and then click Remove.
6. Enter community string information as necessary.

Community. Enter a community string you want to add, or edit the listed community string.

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

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

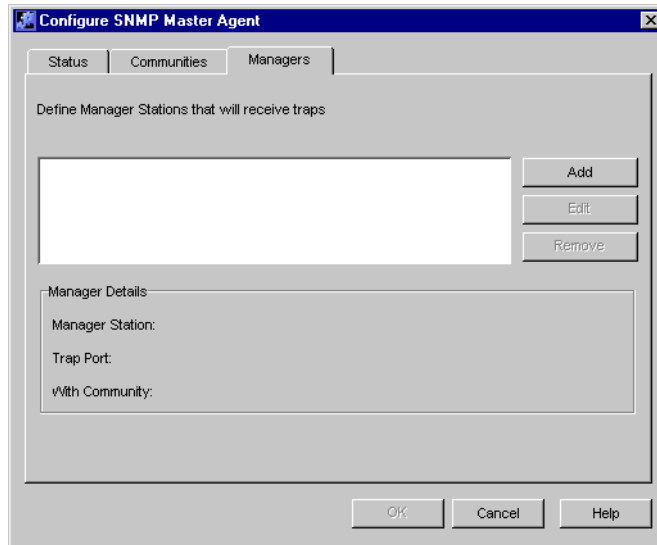
SET only. Choose this option if you want to use this community string only for setting variable values.



7. Click OK.

To Add, Edit, or Remove a Trap Destination

1. In the iPlanet Console navigation tree, select the instance of Administration Server on which the master agent is running.
2. Click Open to open the management window for the server instance.
3. Click the Tasks tab.
4. Click the Configure SNMP Master Agent button, then click Managers.



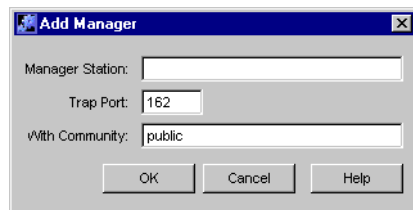
5. Click the appropriate button for the task you are performing.
 - o If you are adding a trap destination, click Add.
 - o If you are editing a trap destination, select it, and then click Edit.

- If you are removing a trap destination, select it, and then click Remove.
- 6. If you are adding or editing a trap destination, enter Manager information as necessary:

Manager Station. Enter a valid system name or an IP address for the network management station.

Trap Port. Enter the port number that the network management station uses to listen for traps. The default is 162.

With Community. Enter the community string you want to use in the trap.



- 7. Click OK.

Manually Configuring the Master Agent

Although you can easily set SNMP master agent parameters through iPlanet Console, you may want to manually add or modify some settings. You can do this by editing the master agent's configuration file. This file is called `CONFIG` and it contains all master agent settings, whether entered manually or through iPlanet Console.

To Configure the Master SNMP Agent Manually

1. Log in as root.
2. Check to see if there is a native agent (`snmpd`) running on port 161.

If a native agent is running, make sure you know which MIB trees it supports and how to restart it, then stop it.
3. Edit the `CONFIG` file located in the `<server-root>/plugins/snmp/magt` directory.
4. (Optional) Define `sysContact` and `sysLocation` variables in the `CONFIG` file.

Instructions for editing the `CONFIG` file and defining the `sysContact` and `sysLocation` variables are detailed below.

Editing the Master Agent Config File

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

```
COMMUNITY      public
                ALLOW ALL OPERATIONS

MANAGER        <your_manager_station_name>
                SEND ALL TRAPS TO PORT 162
                WITH COMMUNITY public
```

Defining `sysContact` and `sysLocation` Variables

You can edit the `CONFIG` file to include initial values for the `sysContact` and `sysLocation` variables (these variables are defined as part of MIB-II, the MIB section of the second version of SNMP). The value for `sysContact` specifies the person in charge of the host system on which the master agent runs. The value for `sysLocation` specifies a physical address where the host machine can be found.

The following example `CONFIG` file defines the `sysContract` and `sysLocation` variables. The strings for the variables in this example are enclosed in quotes. Any string that contains spaces, line breaks, or tabs must be in quotes. Alternatively, you can omit the quotes and specify the value of these whitespace characters in hexadecimal notation.

```
COMMUNITY      public
                ALLOW ALL OPERATIONS

MANAGER        nms2
                SEND ALL TRAPS TO PORT 162
                WITH COMMUNITY public

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

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

Starting the Master Agent on UNIX Systems

Once you have configured the SNMP master agent, you can start it from iPlanet Console or from the command line.

Starting the Agent Using iPlanet Console

iPlanet Console can start the SNMP master agent on the standard port (161) only. If you want to use a non-standard port, see “Starting the Agent From the Command Line” below.

To Start the Master Agent Using iPlanet Console

1. Log in as root.
2. Check to see if there is a native agent (`snmpd`) running on port 161.
If a native agent is running, make sure you know which MIB trees it supports and how to restart it, then stop it.
3. In the iPlanet Console navigation tree, select the instance of Administration Server on which the master agent is running.
4. Click Open to open the management window for the server instance.
5. Click the Tasks tab.
6. Double-click Configure SNMP Master Agent.

7. Click the Start button.

Starting the Agent From the Command Line

If you do not want to start the SNMP master agent from iPlanet Console, you can launch it from the command prompt. If you want to run the agent on a port other than 161, you must modify your `CONFIG` or system services file and then start the agent from the command line.

To Start the Agent on the Standard Port

- Enter the following at the command prompt to start the master agent on port 161:

```
magt CONFIG INIT&
```

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

To Start the Agent on a Non-Standard Port Using the Config File

1. In the `CONFIG` file, specify a transport mapping for each interface over which the master agent listens for SNMP requests from network management stations. Transport mappings allow the master agent to accept connections on both the standard port and a nonstandard port.

The maximum number of concurrent SNMP requests is limited by your target system's limits on the number of open sockets or file descriptors per system process.

Here is an example of a transport mapping entry:

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

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

```
# magt CONFIG INIT&
```

To Start the Agent on a Non-Standard Port Using System Services

- Edit the `/etc/services` file to allow the master agent to accept connections on the standard port as well as on a nonstandard port. For information on editing this file, see your system documentation.

Enabling the Subagent on UNIX Systems

For information on enabling the subagent, see the documentation for your iPlanet or Netscape server. If you need more information, see your system documentation.

Using the Windows NT SNMP Service

Windows NT implements SNMP as a service. Any iPlanet servers that use SNMP communicate directly with this service. iPlanet Administration Server does not perform any SNMP-related tasks on Windows NT. All SNMP-related tasks are handled by the operating system.

To Set Up SNMP on Windows NT Systems

1. Install the SNMP service on your server.
Refer to your Windows NT documentation for instructions.
2. Configure your server software to use SNMP.
For more information, see your server documentation.
3. Click Start, and then choose Settings > Control Panel.
4. Open the Services control panel.
5. Select the SNMP service from the list of services and then click the Start button.
6. Click Close to exit the Services control panel.

Appendixes

Appendix A, “Fortezza”

Appendix B, “Introduction to Public-Key Cryptography”

Appendix C, “Introduction to SSL”

Fortezza

Fortezza is a cryptographic system that combines the use of hardware-based tokens and software-based algorithms to secure electronic information exchange. The US government developed Fortezza to manage sensitive but unclassified information. The information in this appendix applies only to US government agencies and businesses that work with the US government. This appendix contains the following sections:

- How It Works
- How Fortezza Crypto Cards Are Certified
- Fortezza Keys, Certificates, and Encryption
- Enabling Fortezza

How It Works

Fortezza provides a higher level of security than typical encryption systems because it requires three elements:

- A crypto card, which contains a user's unique cryptographic key
- Fortezza encryption algorithms
- Fortezza key management

First, the US government provides your department or agency access to a certificate authority workstation. The workstation itself may or may not be located at your worksite. A certificate authority (CA) representing your department or agency operates the certificate authority workstation. 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 a personal identification number (PIN), the user tells the client to do the following:

- 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 policy approval authority delegates its authority to policy creation authorities (PCAs). These are groups that may represent a branch of the government or a large corporation. Policy creation authorities in turn delegate authority to certificate authorities (CAs).

Certificate authorities are the individuals 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

CAs program Fortezza crypto cards with any combination of key and certificate management approaches and encryption algorithms. Some of these approaches and algorithms are described briefly here. For more information about how keys, certificates, and encryption work in general, see Appendix B, “Introduction to Public-Key Cryptography” and Appendix C, “Introduction to SSL.”

CRLs and CKLs

CAs can provide Certificate revocation lists (CRLs) and compromised key lists (CKLs) to help manage keys and certificates that are stored on Fortezza crypto cards. For information on CRLs and CKLs, see “Managing Certificate Lists,” beginning on page 200.

Encryption Algorithms

CAs can program a number of encryption algorithms into a Fortezza crypto card. This section describes some of the most common 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.

Enabling Fortezza

Enabling Fortezza typically involves installing your card reader, activating SSL, and enabling ciphers.

The following procedure explains how to set up Fortezza on iPlanet Administration Server. Other iPlanet or Netscape 4.x servers may have different setup options and requirements. See your server's documentation for more information.

To Enable Fortezza on Administration Server

1. Install your Fortezza card reader.

See "To Install an External Security Device", on page 186 for more information.

2. Activate SSL.

When prompted to choose ciphers, select the Fortezza ciphers.

See "To Activate SSL on an iPlanet Server or a Netscape 4.x Server", on page 194 for more information.

Introduction to Public-Key Cryptography

Public-key cryptography and related standards and techniques underlie security features of many iPlanet products, including signed and encrypted email, form signing, object signing, single sign-on, and the Secure Sockets Layer (SSL) protocol. This appendix introduces the basic concepts of public-key cryptography. This appendix contains the following sections:

- Internet Security Issues
- Encryption and Decryption
- Digital Signatures
- Certificates and Authentication
- Managing Certificates

For an overview of SSL, see Appendix C, “Introduction to SSL.”

Internet Security Issues

All communication over the Internet uses the Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP allows information to be sent from one computer to another through a variety of intermediate computers and separate networks before it reaches its destination.

The great flexibility of TCP/IP has led to its worldwide acceptance as the basic Internet and intranet communications protocol. At the same time, the fact that TCP/IP allows information to pass through intermediate computers makes it possible for a third party to interfere with communications in the following ways:

- **Eavesdropping.** Information remains intact, but its privacy is compromised. For example, someone could learn your credit card number, record a sensitive conversation, or intercept classified information.
- **Tampering.** Information in transit is changed or replaced and then sent on to the recipient. For example, someone could alter an order for goods or change a person's resume.
- **Impersonation.** Information passes to a person who poses as the intended recipient. Impersonation can take two forms, Spoofing and Misrepresentation.
- **Spoofing.** A person pretends to be someone else. For example, a person can pretend to have the email address `jd@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 misrepresents itself. For example, suppose the site `www.mozilla.com` pretends to be a furniture store when it is really just a site that takes credit-card payments but never sends any goods.

Normally, users of the many cooperating computers that make up the Internet or other networks don't monitor or interfere with the network traffic that continuously passes through their machines. However, many sensitive personal and business communications over the Internet require precautions that address the threats listed above. Fortunately, a set of well-established techniques and standards known as public-key cryptography make it relatively easy to take such precautions.

Public-key cryptography facilitates the following tasks:

- Encryption and decryption allow two communicating parties to disguise information they send to each other. The sender encrypts, or scrambles, information before sending it. The receiver decrypts, or unscrambles, the information after receiving it. While in transit, the encrypted information is unintelligible to an intruder.
- Tamper detection allows the recipient of information to verify that it has not been modified in transit. Any attempt to modify data or substitute a false message for a legitimate one 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.

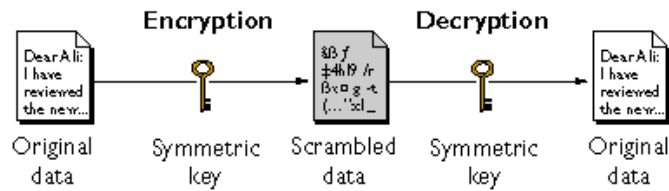
With most modern cryptography, the ability to keep encrypted information secret is based not on the cryptographic algorithm, which is widely known, but on a number called a key that must be used with the algorithm to produce an encrypted result or to decrypt previously encrypted information. Decryption 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 B-1.

Figure B-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.

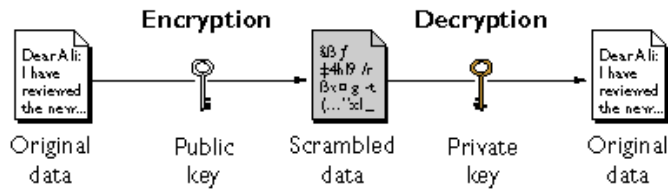
Public-Key Encryption

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,” which begins on page 244.) Data encrypted with your public key can be decrypted only with your private key. Figure B-2 shows a simplified view of the way public-key encryption works.

Figure B-2 Public-Key Encryption



The scheme shown in Figure B-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 B-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 Netscape 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” (beginning on page 242) and subsequent sections describe how this confirmation process works.

Key Length and Encryption Strength

In general, the strength of encryption is related to the difficulty of discovering the key, which in turn depends on both the cipher used and the length of the key. For example, the difficulty of discovering the key for the RSA cipher most commonly used for public-key encryption depends on the difficulty of factoring large numbers, a well-known mathematical problem.

Encryption strength is often described in terms of the size of the keys used to perform the encryption: in general, longer keys provide stronger encryption. Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher. Roughly speaking, 128-bit RC4 encryption is 3×10^{26} times stronger than 40-bit RC4 encryption. (For more information about RC4 and other ciphers used with SSL, see Appendix C, “Introduction to SSL.”)

Different ciphers may require different key lengths to achieve the same level of encryption strength. The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based. Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values. Thus a 128-bit key for use with a symmetric-key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher. This difference explains why the RSA public-key encryption cipher must use a 512-bit key (or longer) to be considered cryptographically strong, whereas symmetric key ciphers can achieve approximately the same level of strength with a 64-bit key. Even this level of strength may be vulnerable to attacks in the near future.

Digital Signatures

Encryption and decryption address the problem of eavesdropping, one of the three Internet security issues mentioned at the beginning of this appendix. But encryption and decryption, by themselves, do not address the other two problems mentioned in “Internet Security Issues” (beginning on page 237): 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,” which begins on page 240, 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 B-3 shows a simplified view of the way a digital signature can be used to validate the integrity of signed data.

Figure B-3 Using a Digital Signature to Validate Data Integrity

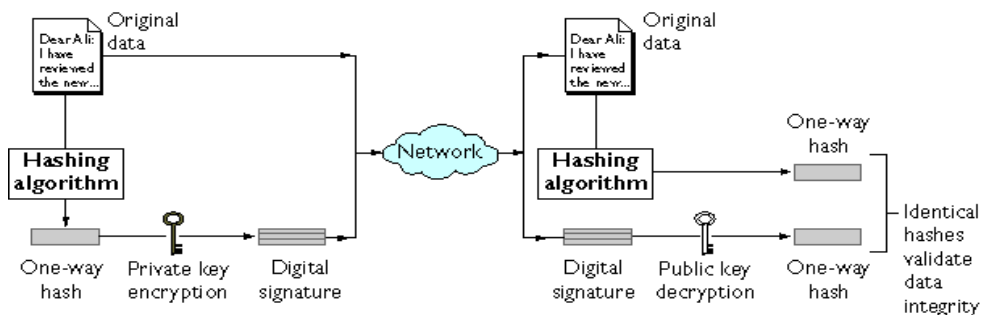


Figure B-3 shows two items transferred to the recipient of some signed data: the original data and the digital signature, which is basically a one-way hash (of the original data) that has been encrypted with the signer’s private key. To validate the integrity of the data, the receiving software first uses the signer’s public key to decrypt the hash. It then uses the same hashing algorithm that generated the original hash to generate a new one-way hash of the same data. (Information about the hashing algorithm used is sent with the digital signature, although this isn’t shown in the figure.) Finally, the receiving software compares the new hash against

the original hash. If the two hashes match, the data has not changed since it was signed. If they don't match, the data may have been tampered with since it was signed, or the signature may have been created with a private key that doesn't correspond to the public key presented by the signer.

If the two hashes match, the recipient can be certain that the public key used to decrypt the digital signature corresponds to the private key used to create the digital signature. Confirming the identity of the signer, however, also requires some way of confirming that the public key really belongs to a particular person or other entity. For a discussion of the way this works, see the next section, "Certificates and Authentication."

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
- Authentication Confirms an Identity
- How Certificates Are Used
- Contents of a Certificate
- How CA Certificates Are Used to Establish Trust

A Certificate Identifies Someone or Something

A *certificate* is an electronic document used to identify an individual, a server, a company, or some other entity. The certificate also associates that identity with a public key. Like a driver's license, a passport, or other commonly used personal IDs, a certificate provides generally recognized proof of a person's identity. Public-key cryptography uses certificates to address the problem of impersonation (see "Internet Security Issues," which begins on page 237).

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 iPlanet Certificate Management System). The methods used to validate an identity vary depending on the policies of a given CA—just as the methods to validate other forms of identification vary depending on who is issuing the ID and the purpose for which it 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," beginning on page 258.

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.

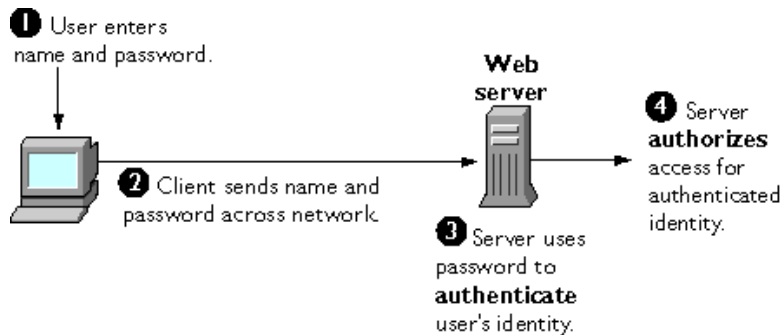
Password-Based Authentication

Figure B-4 shows the basic steps involved in authenticating a client by means of a name and password. Figure B-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 B-4 Using a Password to Authenticate a Client to a Server



These are the steps shown in Figure B-4:

1. In response to an authentication request from the server, the client displays a dialog box requesting the user's name and password for that server. The user must supply a name and password separately for each new server the user wishes to use during a work session.
2. The client sends the name and password across the network, either in the clear or over an encrypted SSL connection.
3. The server looks up the name and password in its local password database and, if they match, accepts them as evidence authenticating the user's identity.
4. The server determines whether the identified user is permitted to access the requested resource, and if so allows the client to access it.

With this arrangement, the user must supply a new password for each server, and the administrator must keep track of the name and password for each user, typically on separate servers.

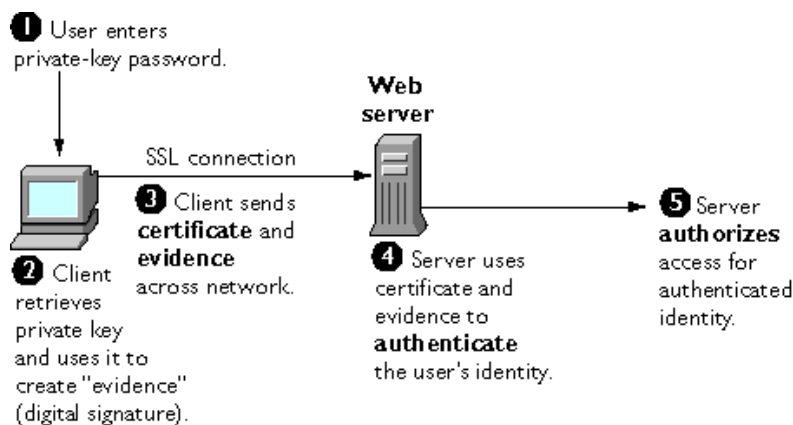
As shown in the next section, one of the advantages of certificate-based authentication is that it can be used to replace the first three steps in Figure B-4 with a mechanism that allows the user to supply just one password (which is not sent across the network) and allows the administrator to control user authentication centrally.

Certificate-Based Authentication

Figure B-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 B-4, Figure B-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 B-5 Using a Certificate to Authenticate a Client to a Server



Unlike the process shown in Figure B-4, the process shown in Figure B-5 requires the use of SSL. Figure B-5 also assumes that the client has a valid certificate that can be used to identify the client to the server. Certificate-based authentication is generally considered preferable to password-based authentication because it is based on what the user has (the private key) as well as what the user knows (the password that protects the private key). However, it's important to note that these two assumptions are true only if unauthorized personnel have not gained access to the user's machine or password, the password for the client software's private key database has been set, and the software is set up to request the password at reasonably frequent intervals.

NOTE Neither password-based authentication nor certificate-based authentication address security issues related to physical access to individual machines or passwords. Public-key cryptography can only verify that a private key used to sign some data corresponds to the public key in a certificate. It is the user's responsibility to protect a machine's physical security and to keep the private-key password secret.

These are the steps shown in Figure B-5:

1. The client software, such as Netscape 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 Appendix C, "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 B-5 to Figure B-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 iPlanet 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,” which begins on page 248, for a description of the way client SSL certificates are used for client authentication. Client SSL certificates can also be used for form signing and as part of a single sign-on solution.

Examples: A bank gives a customer a client SSL certificate that allows the bank's servers to identify that customer and authorize access to the customer's accounts. A company might give a new employee a client SSL certificate that allows the company's servers to identify that employee and authorize access to the company's servers.

- **Server SSL certificates.** Used to identify servers to clients via SSL (server authentication). Server authentication may be used with or without client authentication. Server authentication is a requirement for an encrypted SSL session. For more information, see “SSL Protocol” on page 252.

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,” which begins on page 252). S/MIME certificates can also be used for form signing and as part of a single sign-on solution.

Examples: A company deploys combined S/MIME and SSL certificates solely for the purpose of authenticating employee identities, thus permitting signed email and client SSL authentication but not encrypted email. Another company issues S/MIME certificates solely for the purpose of both signing and encrypting email that deals with sensitive financial or legal matters.

- **Object-signing certificates.** Used to identify signers of Java code, JavaScript scripts, or other signed files. For more information, see “Object Signing,” which begins on page 254.

Example: A software company signs software distributed over the Internet to provide users with some assurance that the software is a legitimate product of that company. Using certificates and digital signatures in this manner can also make it possible for users to identify and control the kind of access downloaded software has to their computers.

- **CA certificates.** Used to identify CAs. Client and server software use CA certificates to determine what other certificates can be trusted. For more information, see “How CA Certificates Are Used to Establish Trust,” which begins on page 258.

Example: The CA certificates stored in Netscape Communicator determine what other certificates that copy of Netscape 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 Netscape Communicator.

The sections that follow describes how certificates are used by iPlanet products.

SSL Protocol

The Secure Sockets Layer (SSL) protocol is a set of rules governing server authentication, client authentication, and encrypted communication between servers and clients. SSL is widely used on the Internet, especially for interactions that involve exchanging confidential information such as credit card numbers.

SSL requires a server SSL certificate, at a minimum. As part of the initial “handshake” process, the server presents its certificate to the client to authenticate the server’s identity. The authentication process uses public-key encryption and digital signatures to confirm that the server is in fact the server it claims to be. Once the server has been authenticated, the client and server use techniques of symmetric-key encryption, which is very fast, to encrypt all the information they exchange for the remainder of the session and to detect any tampering that may have occurred.

Servers may optionally be configured to require client authentication as well as server authentication. In this case, after server authentication is successfully completed, the client must also present its certificate to the server to authenticate the client’s identity before the encrypted SSL session can be established.

For an overview of client authentication over SSL and how it differs from password-based authentication, see “Authentication Confirms an Identity,” which begins on page 245. For more detailed information about SSL, see Appendix C, “Introduction to SSL.”

Signed and Encrypted Email

Some email programs (including Netscape Messenger, which is part of Netscape 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 appendix, 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,” which begins on page 242.)

S/MIME also makes it possible to encrypt email messages. This is also important for some business users. However, using encryption for email requires careful planning. If the recipient of encrypted email messages loses his or her private key and does not have access to a backup copy of the key, for example, the encrypted messages can never be decrypted.

Form Signing

Many kinds of e-commerce require the ability to provide persistent proof that someone has authorized a transaction. Although SSL provides transient client authentication for the duration of an SSL connection, it does not provide persistent authentication for transactions that may occur during that connection. S/MIME provides persistent authentication for email, but e-commerce often involves filling in a form on a web page rather than sending an email message.

The iPlanet 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 Netscape 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 iPlanet utility called the Signature Verification Tool to validate the digital signature.

Single Sign-On

Network users are frequently required to remember multiple passwords for the various services they use. For example, a user might have to type different passwords to log into the network, collect 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 iPlanet products relies on SSL client authentication (see “Certificate-Based Authentication,” which begins on page 248). A user can log in once, using a single password to the local client’s private-key database, and get authenticated access to all SSL-enabled servers that user is authorized to use—without sending any passwords over the network. This approach simplifies access for users, because they don’t need to enter passwords for each new server. It also simplifies network management, since administrators can control access by controlling lists of certificate authorities (CAs) rather than much longer lists of users and passwords.

In addition to using certificates, a complete single-sign on solution must address the need to interoperate with enterprise systems, such as the underlying operating system, that rely on passwords or other forms of authentication.

Object Signing

Netscape Communicator and other Netscape and iPlanet 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 iPlanet products, see *Netscape Object Signing: Establishing Trust for Downloaded Software* at the following URL:

<http://docs.iplanet.com/docs/manuals/signedobj/trust/owp.htm>

Contents of a Certificate

The contents of certificates supported by iPlanet 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 Corporation:

```
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 appendix. For comprehensive information about DNs, see *A String Representation of Distinguished Names* at the following URL:

<http://www.ietf.org/rfc/rfc1485.txt>

A Typical Certificate

Every X.509 certificate consists of two sections:

The data section includes the following information:

- The version number of the X.509 standard supported by the certificate.
- The certificate's serial number. Every certificate issued by a CA has a serial number that is unique among the certificates issued by that CA.
- Information about the user's public key, including the algorithm used and a representation of the key itself.
- The DN of the CA that issued the certificate.
- The period during which the certificate is valid (for example, between 1:00 p.m. on November 15, 1999 and 1:00 p.m. November 15, 2000)
- The DN of the certificate subject (for example, in a client SSL certificate this would be the user's DN), also called the subject name.
- Optional *certificate extensions*, which may provide additional data used by the client or server. For example, the certificate type extension indicates the type of certificate—that is, whether it is a client SSL certificate, a server SSL certificate, a certificate for signing email, and so on. Certificate extensions can also be used for a variety of other purposes.

The signature section includes the following information:

- The cryptographic algorithm, or cipher, used by the issuing CA to create its own digital signature. For more information about ciphers, see Appendix C, "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
MA8GA1UEChMITmV0c2NhcgUxFTATBgNVBAsTDFNlcHJpeWEncyBDQTAeFw05NzEw
MTgwMTM2MjVaFw05OTEwMTgwMTM2MjVaMEgxCzAJBgNVBAYTA1VTMREwDwYDVQQK
EwhOZXRzY2FwZTENMA8GA1UECxEUHViczEXMBUGA1UEAxMOU3Vwcmcl5YSBtaGV0
dHkwZz8wDQYJKoZIhvcNAQEFBQADgY0AMIGJAoGBAMr6eZiPGfjX3uRjgEjmKiQG
7SdATYazBcABu1AVyd7chRkiQ31FbXFOGD3wNktbf6hRo6EAmM5/R1AskzZ8AW7L
iQZBcrXpc0k4du+2Q6xJu2MPm/8WKuMOnTuvzpo+SGXelmHVChEqooCwfdiZywyZ
NMmrJgaoMa2MS6pUkfQVAgMBAAGjNjA0MBEGCWCsAGG+EIBAQQEAWIAGDAfBgNV
HSMEGDAWgBTy8gZzkBhHufWJM1oxeuZc+zYmyTANBgkqhkiG9w0BAQQFAAQBt
I6/z07Z635DfzX4XbAFpjlRl/AYwQzTSYx8GfcNAqCqCwaSDKvsuj/vwbf91o3j3
UkdGYpcd2cYRCgKi4MwqdWyLtpuHAH18hHZ5uvi00mJYw8W2wUOsY0RC/a/IDy84
hW3WWehBUqVK5SY4/zJ4oTjx7dwNMdGwbWfprqjdlA==
-----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 iPlanet Certificate Management System). A list of third-party certificate authorities is available at “Certificate Authority Services” (<https://certs.netscape.com/client.html>).

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 explain how certificate hierarchies and certificate chains determine what certificates software can trust.

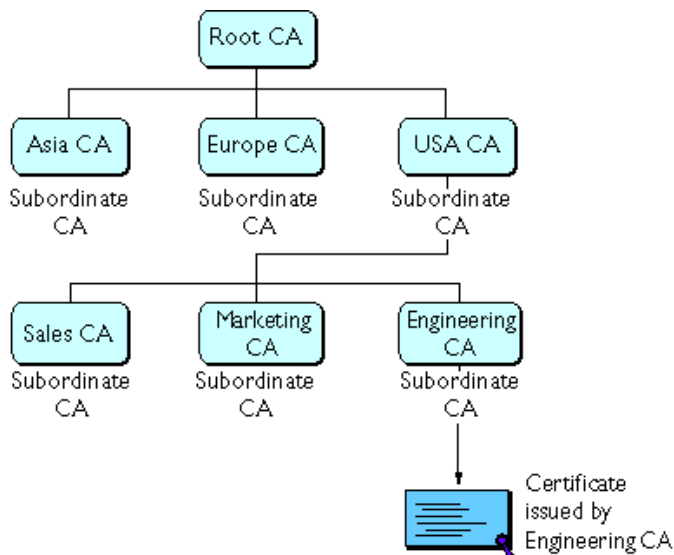
- CA Hierarchies
- Certificate Chains
- Verifying a Certificate Chain

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 B-6.

Figure B-6 Example of a Hierarchy of Certificate Authorities



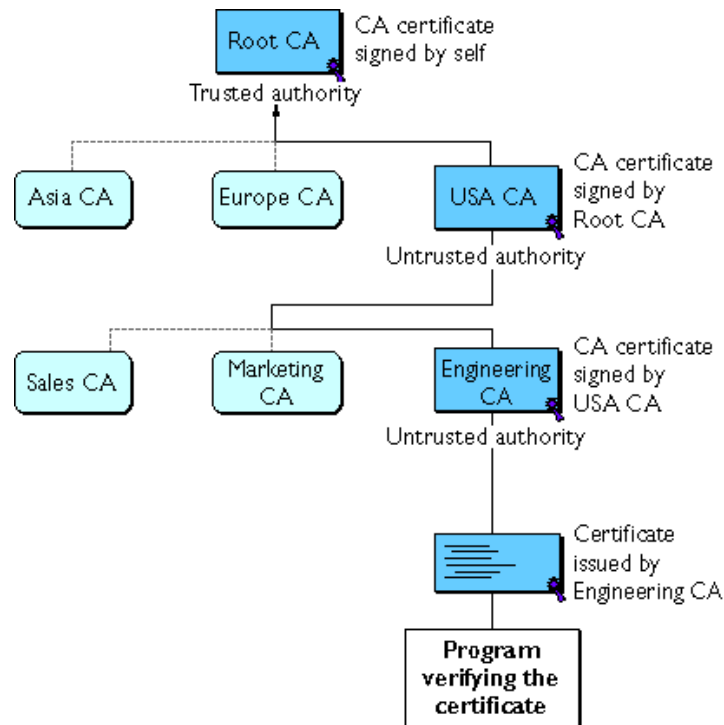
In this model, the root CA is at the top of the hierarchy. The root CA's certificate is a *self-signed certificate*: that is, the certificate is digitally signed by the same entity—the root CA—that the certificate identifies. The CAs that are directly subordinate to the root CA have CA certificates signed 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 B-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 B-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 B-6).

Figure B-7 Example of a Certificate Chain



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 B-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 B-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. iPlanet 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 iPlanet certificate type extension, and chain verification returns to step 1 to start again, but with this new certificate. Figure B-8 presents an example of this process.

Figure B-8 Verifying a Certificate Chain All the Way to the Root CA

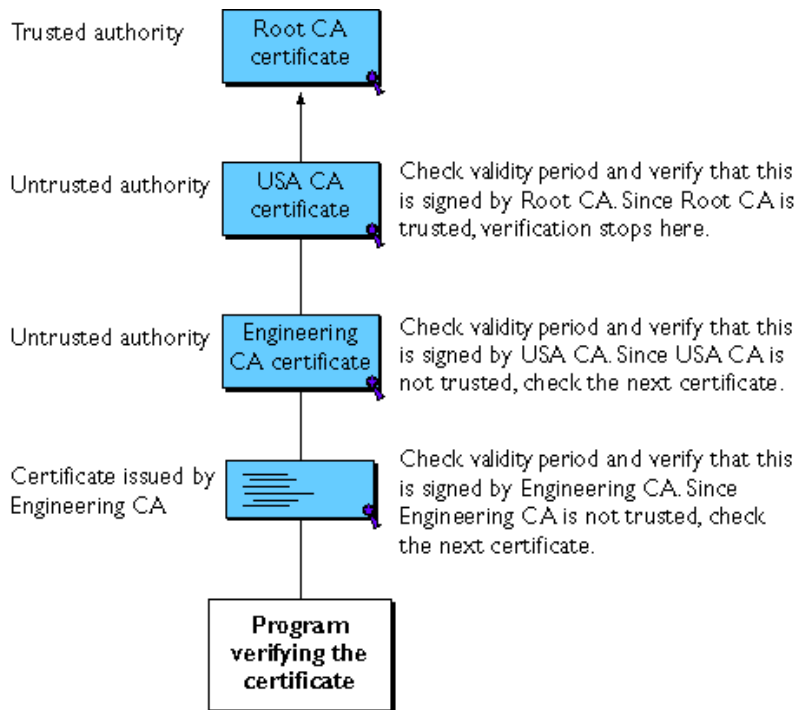
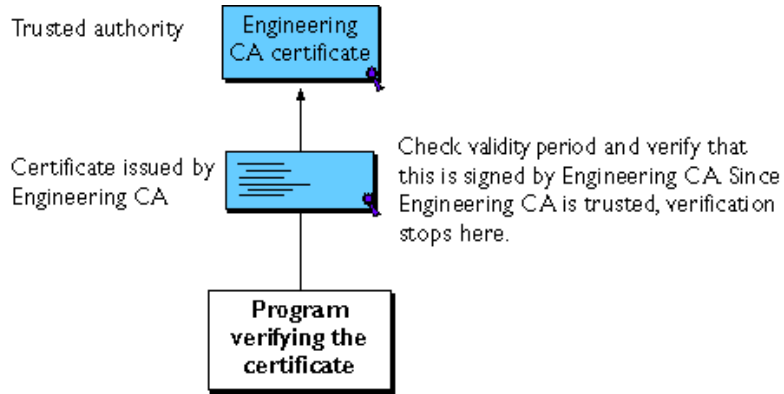


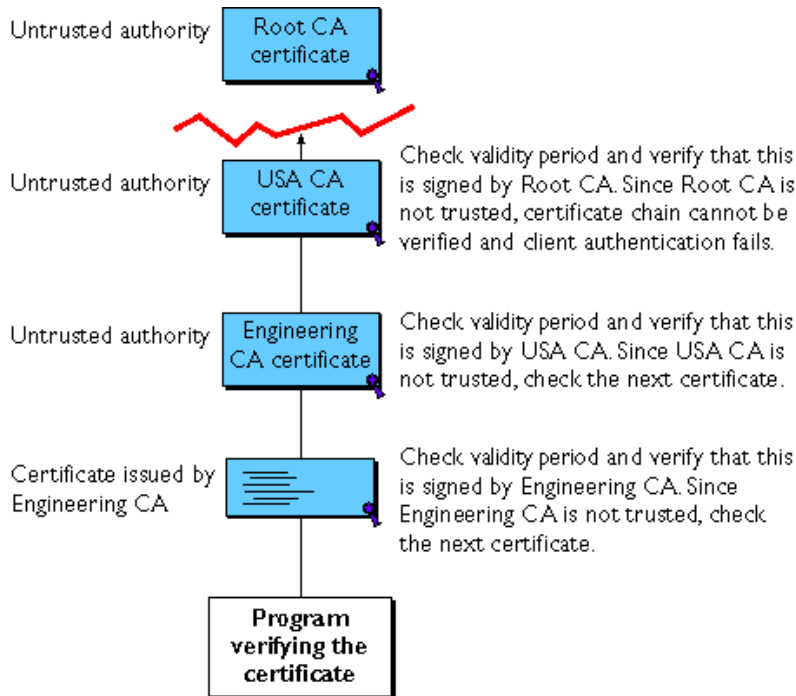
Figure B-8 shows what happens when only Root CA is included in the verifier's local database. If a certificate for one of the intermediate CAs shown in Figure B-8, such as Engineering CA, is found in the verifier's local database, verification stops with that certificate, as shown in Figure B-9.

Figure B-9 Verifying a Certificate Chain to an Intermediate CA



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 B-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 B-10 A Certificate Chain That Can't Be Verified



For general information about the way digital signatures work, see “Digital Signatures,” which begins on page 242. For a more detailed description of the signature verification process in the context of SSL client and server authentication, see Appendix C, “Introduction to SSL.”

Managing Certificates

The set of standards and services that facilitate the use of public-key cryptography and X.509 v3 certificates in a network environment is called the *public key infrastructure* (PKI). PKI management is complex topic beyond the scope of this appendix. The sections that follow introduce some of the specific certificate management issues addressed by iPlanet 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.

iPlanet Certificate Management System allows an organization to set up its own certificate authority and issue certificates.

Issuing certificates is one of several managements tasks that can be handled by separate Registration Authorities.

Certificates and the LDAP Directory

The Lightweight Directory Access Protocol (LDAP) for accessing directory services supports great flexibility in the management of certificates within an organization. System administrators can store much of the information required to manage certificates in an LDAP-compliant directory. For example, a CA can use information in a directory to prepopulate a certificate with a new employee's legal name and other information. The CA can leverage directory information in other ways to issue certificates one at a time or in bulk, using a range of different identification techniques depending on the security policies of a given organization. Other routine management tasks, such as key management and renewing and revoking certificates, can be partially or fully automated with the aid of the directory.

Information stored in the directory can also be used with certificates to control access to various network resources by different users or groups. Issuing certificates and other certificate management tasks can thus be an integral part of user and group management.

In general, high-performance directory services are an essential ingredient of any certificate management strategy. iPlanet Directory Server is fully integrated with iPlanet Certificate Management System to provide a comprehensive certificate management solution.

Key Management

Before a certificate can be issued, the public key it contains and the corresponding private key must be generated. Sometimes it may be useful to issue a single person one certificate and key pair for signing operations, and another certificate and key pair for encryption operations. Separate signing and encryption certificates make it possible to keep the private signing key on the local machine only, thus providing maximum nonrepudiation, and to back up the private encryption key in some central location where it can be retrieved in case the user loses the original key or leaves the company.

Keys can be generated by client software or generated centrally by the CA and distributed to users via an LDAP directory. There are trade-offs involved in choosing between local and centralized key generation. For example, local key generation provides maximum nonrepudiation, but may involve more participation by the user in the issuing process. Flexible key management capabilities are essential for most organizations.

Key recovery, or the ability to retrieve backups of encryption keys under carefully defined conditions, can be a crucial part of certificate management (depending on how an organization uses certificates). Key recovery schemes usually involve an *m of n* mechanism: for example, *m* of *n* managers within an organization might have to agree, and each contribute a special code or key of their own, before a particular person's encryption key can be recovered. This kind of mechanism ensures that several authorized personnel must agree before an encryption key can be recovered.

Renewing and Revoking Certificates

Like a driver's license, a certificate specifies a period of time during which it is valid. Attempts to use a certificate for authentication before or after its validity period 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 serviced.

As previously discussed, the means used by different CAs to verify an identity before issuing a certificate can vary widely, depending on the organization and the purpose for which the certificate 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.

Introduction to SSL

This appendix 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. This appendix contains the following sections:

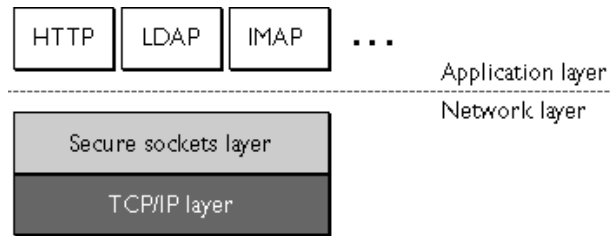
- The SSL Protocol
- Ciphers Used With SSL
- The SSL Handshake

The new Internet Engineering Task Force (IETF) standard protocol called Transport Layer Security (TLS) is based on SSL. The details of the protocol are available in Request For Comments (RFC): 2246, *The TLS Protocol Version 1.0*. Some iPlanet and Netscape products already support TLS. Most other iPlanet products plan to support the protocol in future versions.

This appendix is primarily intended for administrators of iPlanet and Netscape server products, but the information it contains may also be useful for developers of applications that support SSL. The appendix assumes that you are familiar with the basic concepts of public-key cryptography, as summarized in Appendix B, "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 C-1 Where SSL Runs

The SSL protocol runs above TCP/IP and below higher-level protocols such as HTTP or IMAP. It uses TCP/IP on behalf of the higher-level protocols, and in the process allows an SSL-enabled server to authenticate itself to an SSL-enabled client, allows the client to authenticate itself to the server, and allows both machines to establish an encrypted connection.

These capabilities address fundamental concerns about communication over the Internet and other TCP/IP networks:

- SSL server authentication allows a user to confirm a server's identity. SSL-enabled client software can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs. This confirmation might be important if the user, for example, is sending a credit card number over the network and wants to check the receiving server's identity.
- SSL client authentication allows a server to confirm a user's identity. Using the same techniques as those used for server authentication, SSL-enabled server software can check that a client's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the server's list of trusted CAs. This confirmation might be important if the server, for example, is a bank sending confidential financial information to a customer and wants to check the recipient's identity.
- An encrypted SSL connection requires all information sent between a client and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality. Confidentiality is important for both parties to any private transaction. In addition, all data sent over an encrypted SSL connection is protected with a mechanism for detecting tampering—that is, for automatically determining whether the data has been altered in transit.

The SSL protocol includes two sub-protocols: the SSL record protocol and the SSL handshake protocol. The SSL record protocol defines the format used to transmit data. The SSL handshake protocol involves using the SSL record protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when they first establish an SSL connection. This exchange of messages is designed to facilitate the following actions:

- Authenticate the server to the client.
- Allow the client and server to select the cryptographic algorithms, or ciphers, that they both support.
- Optionally authenticate the client to the server.
- Use public-key encryption techniques to generate shared secrets.
- Establish an encrypted SSL connection.

For more information about the handshake process, see “The SSL Handshake,” which begins on page 275.

Ciphers Used With SSL

The SSL protocol supports the use of a variety of different cryptographic algorithms, or ciphers, for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. Clients and servers may support different cipher suites, or sets of ciphers, depending on factors such as the version of SSL they support, and company policies regarding acceptable encryption strength. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suites they 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.

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

These are the cipher suites for SSL 3.0:

- RC4 with 128-bit encryption and MD5 message authentication
- RC4 with 40-bit encryption and MD5 message authentication
- RC2 with 40-bit encryption and MD5 message authentication
- No encryption, MD5 message authentication only

See Table C-1 for details.

Cipher Suites With RSA Key Exchange

Table C-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 C-1 Cipher Suites Supported by the SSL Protocol That Use the RSA Key-Exchange Algorithm

Strength Category and Recommended Use	Cipher Suites
<p>Strongest Cipher Suite</p> <p>This cipher suite is appropriate for banks and other institutions that handle highly sensitive data.</p> <p>iPlanet Console does not support this cipher suite.</p>	<p>Triple DES With 168-Bit Encryption and SHA-1 Message Authentication</p> <p>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 $3.7 * 10^{50}$.</p> <p>This cipher suite is FIPS-compliant.</p> <p>Both SSL 2.0 and SSL 3.0 support this cipher suite.</p>

Table C-1 Cipher Suites Supported by the SSL Protocol That Use the RSA Key-Exchange Algorithm

Strength Category and Recommended Use	Cipher Suites
<p>Strong Cipher Suites</p> <p>These cipher suites support encryption that is strong enough for most business or government needs.</p>	<p>RC4 With 128-Bit Encryption and MD5 Message Authentication</p> <p>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.</p> <p>Both SSL 2.0 and SSL 3.0 support this cipher suite.</p> <p>iPlanet Console supports only the SSL 3.0 version of this cipher suite.</p> <p>RC2 With 128-Bit Encryption and MD5 Message Authentication</p> <p>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.</p> <p>This cipher suite is supported by SSL 2.0 but not by SSL 3.0.</p> <p>iPlanet Console does not support this cipher suite.</p> <p>DES With 56-Bit Encryption and SHA-1 Message Authentication</p> <p>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.</p> <p>This cipher suite is FIPS-compliant.</p> <p>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.</p> <p>iPlanet Console does not support this cipher suite.</p>
<p>Less Strong Cipher Suites</p> <p>These cipher suites are not as strong as those listed above, but are widely used.¹</p>	<p>RC4 With 40-Bit Encryption and MD5 Message Authentication</p> <p>RC4 40-bit encryption permits approximately $1.1 * 10^{12}$ (a trillion) possible keys. RC4 ciphers are the fastest of the supported ciphers.</p> <p>Both SSL 2.0 and SSL 3.0 support this cipher.</p> <p>iPlanet Console supports only the SSL 3.0 version of this cipher suite.</p>

Table C-1 Cipher Suites Supported by the SSL Protocol That Use the RSA Key-Exchange Algorithm

Strength Category and Recommended Use	Cipher Suites
Weakest Cipher Suite	RC2 With 40-Bit Encryption and MD5 Message Authentication
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.	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. iPlanet Console supports only the SSL 3.0 version of this cipher suite.
	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.

1. Note that for RC4 and RC2 ciphers, the phrase "40-bit encryption" means the keys are still 128 bits long, but only 40 bits have cryptographic significance.

Fortezza Cipher Suites

Table C-2 lists additional cipher suites supported by most 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. However, Fortezza is not exclusive to the the U.S. government. 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 C-2 Cipher Suites Supported by Netscape Products When Using Fortezza for SSL 3.0

Strength Category and Recommended Use	Cipher Suites
<p>Strong Fortezza Cipher Suites</p> <p>These cipher suites support encryption that is strong enough for most business or government needs.</p> <p>iPlanet Console does not support these cipher suites.</p>	<p>RC4 With 128-bit Encryption and SHA-1 Message Authentication</p> <p>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.</p> <p>This cipher suite is supported by SSL 3.0 but not by SSL 2.0.</p> <p>RC4 With SKIPJACK 80-Bit Encryption and SHA-1 Message Authentication</p> <p>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.</p> <p>This cipher suite is supported by SSL 3.0 but not by SSL 2.0.</p>
<p>Weakest Fortezza Cipher Suite</p> <p>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.</p> <p>iPlanet Console does not support these cipher suites.</p>	<p>No Encryption, SHA-1 Message Authentication Only</p> <p>This cipher uses SHA-1 message authentication to detect tampering.</p> <p>This cipher suite is supported by SSL 3.0 but not by SSL 2.0.</p>

The SSL Handshake

The SSL protocol uses a combination of public-key and symmetric key encryption. Symmetric key encryption is much faster than public-key encryption, but public-key encryption provides better authentication techniques. An SSL session always begins with an exchange of messages called the *SSL handshake*. The handshake allows the server to authenticate itself to the client using public-key

techniques, then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. Optionally, the handshake also allows the client to authenticate itself to the server.

The exact programmatic details of the messages exchanged during the SSL handshake are beyond the scope of this appendix. However, the steps involved can be summarized as follows (assuming the use of the cipher suites listed in “Cipher Suites With RSA Key Exchange,” which begins on page 272):

1. The client sends the server the client’s SSL version number, cipher settings, randomly generated data, and other information the server needs to communicate with the client using SSL.
2. The server sends the client the server’s SSL version number, cipher settings, randomly generated data, and other information the client needs to communicate with the server over SSL. The server also sends its own certificate and, if the client is requesting a server resource that requires client authentication, requests the client’s certificate.
3. The client uses some of the information sent by the server to authenticate the server (for details, see “Server Authentication,” which begins on page 278). If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client goes on to step 4.
4. Using all data generated in the handshake so far, the client (with the cooperation of the server, depending on the cipher being used) creates the premaster secret for the session, encrypts it with the server’s public key (obtained from the server’s certificate, sent in step 2), and sends the encrypted premaster secret to the server.
5. If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case the client sends both the signed data and the client’s own certificate to the server along with the encrypted premaster secret.
6. If the server has requested client authentication, the server attempts to authenticate the client (for details, see “Client Authentication,” which begins on page 280). 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, iPlanet and 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 is terminated.
- In the case of client authentication, the client encrypts some random data with the client's private key—that is, it creates a digital signature. The public key in the client's certificate can correctly validate the digital signature only if the corresponding private key was used. Otherwise, the server cannot validate the digital signature and the session is terminated.

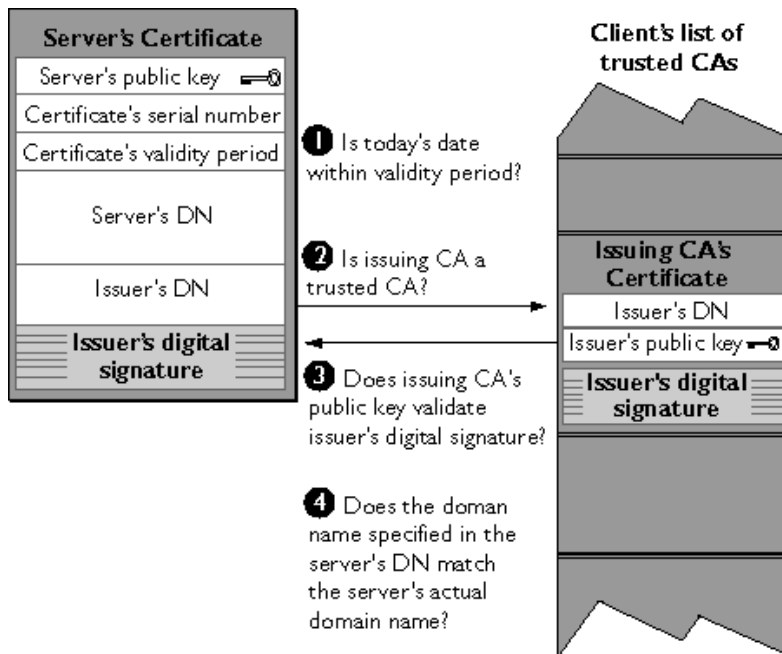
The sections that follow provide more details on server authentication and client authentication.

Server Authentication

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," which begins on page 275, 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 C-2. 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 C-2 Authentication of a Client Certificate



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 C-3. 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" on page 259 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," which begins on page 280.

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,” which begins on page 275. 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 necessary to protect against a man-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 exchanged at the beginning of the SSL handshake is actually encrypted with the rogue program’s public key or private key, rather than the client’s or server’s real keys. The rogue program ends up establishing one set of session keys for use with the real server, and a different set of session keys for use with the client. This allows the rogue program not only to read all the data that flows between the client and the real server, but also to change the data without being deleted. Therefore, it is extremely important for the client to check that the domain name in the server certificate corresponds to the domain name of the server with which a client is attempting to communicate—in addition to checking the validity of the certificate by performing the other steps described in “Server Authentication” on page 278.

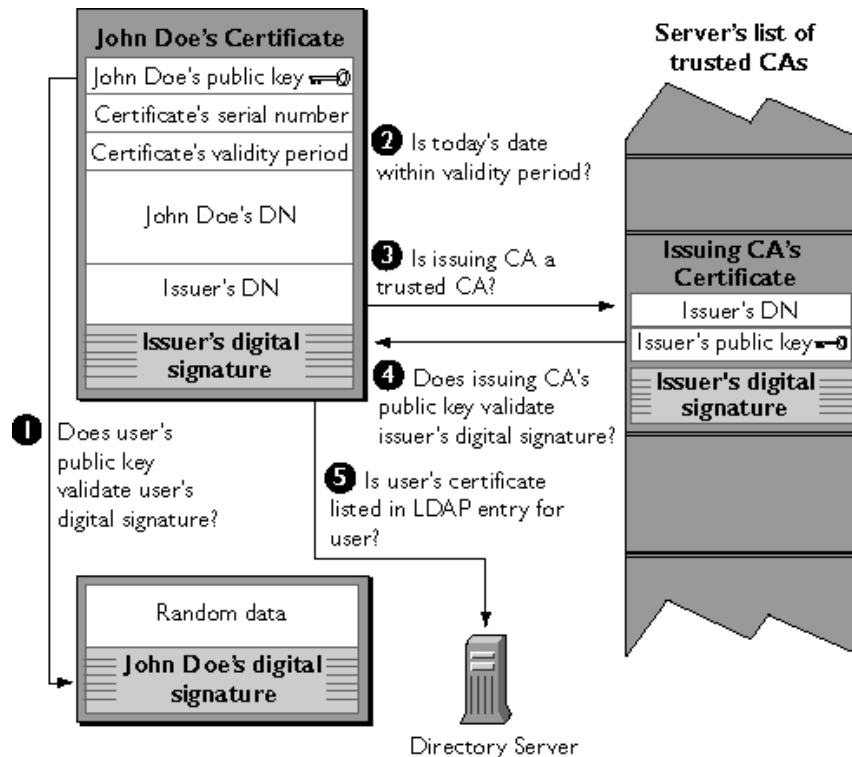
Client Authentication

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,” which begins on page 275), 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 C-3. Although the fifth question is not part of the SSL protocol, iPlanet and 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 C-3 Authentication and Verification of a Client Certificate



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 C-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" on page 259 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. iPlanet and Netscape servers may optionally be configured to perform step 5 before step 6.
5. **Is the user's certificate listed in the LDAP entry for the user?** This optional step provides one way for a system administrator to revoke a user's certificate even if it passes the tests in all the other steps. The iPlanet Certificate Management System can automatically remove a revoked certificate from the

user's entry in the LDAP directory. All servers that are set up to perform this step 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.

The SSL Handshake

Glossary

access control The process of controlling who is allowed to do what to a server, onscreen element, task, or directory entry. See also **access control instruction (ACI)**, **access control list (ACL)**.

access control instruction (ACI) A rule that permits or restricts access to a server, onscreen element, task, or directory entry.

access control list (ACL) A collection of ACIs used to perform complex authorization procedures.

administration domain A collection of host systems and servers that share the same user directory.

Administration Server An HTTP server that acts as the back end to iPlanet Console. A single instance of Administration Server manages operation requests from all servers installed in a server group.

Administration Server Administrator The user who can log in to iPlanet Console even when an instance of Administration Server is not connected to an instance of Directory Server. The Administration Server Administrator is not in the user directory, but is created and stored locally (on the server machine) during installation of Administration Server.

administrator A user who manages and configures servers.

attribute A descriptive aspect of a directory entry. Consists of a label, an attribute type, and one or more attribute values. For example, a user entry might have an attribute called `telephoneNumber` that contains the value `(555)555-5555`.

authentication Assurance that a party to a computerized transaction is not an impostor. Authentication typically involves the use of a password, certificate, PIN, or other information that can be used to validate identity over a computer network. See also **certificate-based authentication**, **client authentication**, **password-based authentication**, **server authentication**.

bind DN A user ID, in the form of a distinguished name (DN), used with a password to authenticate to Netscape or iPlanet Directory Server.

browser Software, such as Netscape Navigator, used to request and view World Wide Web material stored as HTML files. The browser uses the HTTP protocol to communicate with the host server. Also known as a client program.

CA See **certificate authority (CA)**.

CA certificate A certificate that identifies a certificate authority. See also **certificate authority (CA)**, **root CA**.

CA hierarchy A hierarchy of CAs in which a root CA delegates the authority to issue certificates to subordinate CAs. Subordinate CAs can also expand the hierarchy by delegating issuing status to other CAs. See also **certificate authority (CA)**, **root CA**.

certificate Digital data that specifies the name of an individual, company, or other entity and certifies that a public key, which is also included in the certificate, belongs to that entity. A certificate is issued and digitally signed by a certificate authority (CA). A certificate's validity can be verified by checking the CA's digital signature using the techniques of public-key cryptography.

certificate-based authentication Authentication using certificates. See **server authentication**, **client authentication**.

certificate authority (CA) A trusted issuer of certificates. CAs are responsible for verifying the identity of the person or entity that a certificate represents. A CA also renews and revokes certificates and generates CRLs. Certificate authorities can be independent third parties (such as those listed at <https://certs.netscape.com/client.html>) or a person or organization using certificate-issuing server software.

certificate authority workstation A computer used to program Fortezza crypto cards.

certificate chain A hierarchical series of certificates signed by successive certificate authorities. A certificate chain contains a CA certificate that identifies a certificate authority (CA) and that is used to sign certificates issued by that authority. This CA certificate can in turn be signed by the CA certificate of a parent CA, and so on up to a root CA.

certificate extensions Data that is included with a certificate, but that is not part of the standard set of certificate information.

certificate group A group of users who have a certificate containing a common attribute. For example, suppose a certificate is created for all users who have the attributes `ou=Engineering, ou=Anytown`. An administrator can create an “Anytown Engineers” certificate group that grants special access to users whose certificates contain these attributes. When a user presents the server with a certificate containing these attributes, he is identified as part of the Anytown Engineers certificate group and is then granted appropriate access rights.

certificate revocation list (CRL) A list of revoked certificates generated and signed by a certificate authority (CA).

cipher A set of rules or directions used to perform cryptographic operations such as encryption and decryption.

cipher suite Sets of ciphers.

CKL See **compromised key list (CKL)**.

client authentication The process of identifying a client to a server using a name and password or a certificate and some digitally signed data. See also **certificate-based authentication, password-based authentication, server authentication**.

client program See **browser**.

cloning The act of copying the configuration data in one server to multiple servers of the same type.

compromised key list (CKL) A list of keys that have been compromised or otherwise tampered with.

Configuration Administrator The person who can manage all resources in the iPlanet Console navigation tree.

Configuration Administrators group A static group whose members have unrestricted access to the configuration directory. The group is stored in the configuration directory under the following DN:

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

configuration directory Typically, a subtree of a directory containing application and server configuration information. In large deployments, the configuration directory can be a separate instance of Directory Server.

connection restrictions Rules that specify which hosts are allowed to connect to an instance of Administration Server.

CRL See **certificate revocation list (CRL)**.

crypto card See **Fortezza crypto card**.

cryptographic algorithm See **cipher**.

decryption The unscrambling of data that has been encrypted. See also **encryption**.

Directory Server gateway A collection of HTML forms that allows a browser to perform LDAP client functions, such as querying and accessing an instance of Directory Server.

distinguished name (DN) String representation of an entry's location in an LDAP directory. Every distinguished name is unique.

DN See **distinguished name (DN)**.

DNS Domain Name System. The system used by machines on a network to associate standard IP addresses (such as 172.17.66.98) with host names (such as www.iplanet.com). Machines typically get the IP address for a hostname from a DNS server, or they look it up in tables maintained on their systems.

dynamic group A group into which members are automatically added based on their DN attributes.

eavesdropping Surreptitious interception of information sent over a network by an entity for which the information is not intended.

encryption The process of scrambling information in a way that disguises its meaning. See also **decryption**.

external security device A key-pair and certificate database stored in an external device such as a smart card.

failover support The ability to check multiple instances of Directory Server when authenticating a user. This is useful when the instance of Directory Server containing your primary user directory is not accessible.

Fortezza A cryptographic system, developed by the US government, that combines the use of hardware-based tokens and software-based algorithms to secure electronic information exchange.

Fortezza crypto card A PCMCIA card that contains a user's unique key, as well as certificate management approaches and encryption algorithms used by Fortezza.

gateway See **Directory Server gateway**.

group A collection of users who share a common attribute.

hostname A name for a machine in the form `machine.domain.dom`, which is translated into an IP address. For example, `www.iplanet.com` is the machine `www` in the subdomain `iplanet` and `com` domain.

HTML Hypertext Markup Language. The formatting language used for documents on the World Wide Web. HTML files are plain text files with formatting codes that tell browsers such as Netscape Navigator how to display text, position graphics and form items, and display links to other pages.

HTTP Hypertext Transfer Protocol. The method for exchanging information between HTTP servers and clients.

impersonation The act of posing as the intended recipient of information sent over a network. Impersonation can take two forms: spoofing and misrepresentation.

information panel The right-hand side of the "Servers and Applications" tab in the main iPlanet Console window. Displays detailed information about a selected resource.

instance See **server instance**.

internal security device A key-pair and a certificate database stored in a software file on a host computer.

IP address Internet Protocol address. A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 172.17.66.98).

IP spoofing The forgery of client IP addresses.

iPlanet Console The Java application used to manage iPlanet and Netscape servers as well as entries in the user directory.

JAR file A compressed collection of Java class files.

JAR information file A text file containing special scripting instructions. This file is used by `modutil` when handling JAR files.

key (1) A number used by a cryptographic algorithm to encrypt or decrypt data. See also **public key** and **private key**. (2) Predefined commands and options that `modutil` interprets.

key and certificate database A collection of keys and certificates used by a server instance or client.

key recovery The ability to retrieve backups of encryption keys under carefully defined conditions.

LDAP See **Lightweight Directory Access Protocol (LDAP)**.

LDAP Data Interchange Format See **LDIF**.

LDIF LDAP Data Interchange Format. Format used to represent Directory Server entries in text form.

Lightweight Directory Access Protocol (LDAP) A subset of the X.500 protocol, LDAP is a communication standard used for storing and accessing information in directories.

managed devices A piece of hardware or software that is controlled over SNMP.

managed object Configuration and management settings that can be read and changed by an SNMP master agent.

management information base See **MIB**.

master agent See **SNMP master agent**.

member A directory entry that is part of a group. For instance, in a dynamic group called Western Sales, members might include all users whose directory entries contain the RDN `ou=Western Sales`.

MIB Management Information Base. A tree-like hierarchy that defines managed objects.

migration The act of importing settings from one version of a server to a later version of the same server.

misrepresentation The presentation of an entity as a person or organization that it is not. For example, a web site might pretend to be a furniture store when it is really just a site that takes credit-card payments but never sends any goods. Misrepresentation is one form of impersonation. See also **spoofing**.

modutil The Security Module Database Tool. A command-line utility for managing PKCS #11 module information stored in `secmod.db` files or hardware tokens.

native agent An SNMP master agent that is built into a version of the UNIX operating system.

navigation tree The graphical representation in iPlanet Console of a network topology. A navigation tree contains all resources that are registered in a configuration directory.

network management application An application that shows information about managed devices.

network management station (NMS) The machine used to monitor and configure managed devices.

network topology See **topology**.

NMS See **network management station (NMS)**.

nonrepudiation The inability of a sender of information to claim that the information was never sent. A digital signature provides one form of nonrepudiation.

object class A definition of a type of directory entry. An object class includes definitions of the attributes that are contained in a directory entry.

organizational unit A directory entry that can include a number of groups. Usually represents a division, department, or other discrete business group.

ou Abbreviation for organizational unit in a distinguished name (DN).

password-based authentication Authentication using passwords.

PKCS #11 The public-key cryptography standard that governs cryptographic security devices such as smart cards.

PKCS #11 module A driver for a device that provides cryptographic services such as encryption and decryption via the PKCS #11 interface. A PKCS #11 module can be implemented in either hardware or software, and always contains one or more slots. Each of these slots, which can be implemented physically in hardware or conceptually in software, can contain a security device. iPlanet Console includes a built-in software PKCS #11 module.

port number A way to identify a specific process to which a network message is to be forwarded when it arrives at a server.

POSIX Portable Operating System Interface for UNIX, is a standard for the interface between UNIX and application programs.

private key One of a pair of keys used in public-key cryptography. The private key is kept secret and is used to decrypt data encrypted with the corresponding public key.

protocol A set of rules that describes how devices on a network exchange information.

public key One of a pair of keys used in public-key cryptography. The public key is distributed freely and published as part of a certificate. It is typically used to encrypt data sent to the public key's owner, who then decrypts the data with the corresponding private key.

public-key encryption A set of encryption techniques that use a public key and a private key.

public-key infrastructure (PKI) The standards and services that facilitate the use of public-key encryption and certificates in a networked environment.

RDN See **relative distinguished name (RDN)**.

RDN Keyword An abbreviation that is part of a distinguished name.

registration authority (RA) An entity that receives and authenticates certificate requests, and then forwards them to a CA.

relative distinguished name (RDN) The name of a directory entry, before the entry's ancestors have been appended to the string to form the full distinguished name.

resource An object in an iPlanet topology. Examples of resources include administration domains, hosts, and server instances.

RFC Request For Comments. Procedures or standards documents submitted to the Internet community. Readers can send comments on the technologies before they become accepted standards.

root CA The certificate authority (CA) with a self-signed certificate at the top of a certificate chain. See also **CA certificate**.

schema Definitions describing what types of information can be stored as entries in the directory. When information that does not match the schema is stored in the directory, clients attempting to access the directory may be unable to display the proper results.

schema checking Ensures that new or modified directory entries conform to the defined schema. Schema checking is turned on by default; users will receive an error if they try to save an entry that does not conform to the schema.

Secure Sockets Layer (SSL) A protocol that allows mutual authentication between a client and server for the purpose of establishing an authenticated and encrypted connection. SSL runs above TCP/IP and below HTTP, LDAP, IMAP, NNTP, and other high-level network protocols. See also **authentication**, **encryption**.

security device A hardware or software device that is associated with a slot in a PKCS #11 module. It provides cryptographic services and optionally stores certificates and keys. See also **internal security device**, **external security device**.

self-signed certificate A certificate that is digitally signed by the same entity that the certificate identifies.

server Instances of server software that provide specific services such as a directory database, and messaging, and publishing.

server authentication The process of identifying a server to a client. See also **certificate-based authentication**.

server certificate A single certificate, associated only with your server, that identifies your server to clients. See also **certificate**.

server certificate chain A collection of certificates automatically generated for you by your company's internal certificate server or a known CA. The certificates in a chain trace back to the original CA, providing proof of identity. See also **certificate chain**.

server group The servers in a server root that are managed by a single instance of iPlanet Administration Server.

server instance An individual server that shares a machine with other servers of the same type. Instances are virtual servers that share a single installation of a product. For example, if an ISP handles mail for siroe.com, it can install iPlanet Messaging server and create a single instance. If the ISP begins handling mail for another domain, it can create a second instance of Messaging server on the same computer without installing any additional software.

server root A folder that holds server programs and configuration, maintenance, and information files. The servers in a server root make up a server group.

session See **SSL session**.

session key Symmetric keys used to encrypt and decrypt information exchanged during an SSL session and to verify its integrity.

Simple Network Management Protocol (SNMP) A protocol used to exchange data about network activity. SNMP defines a standard method of communication used to manage products from different vendors..

single sign-on The capability for a user to log in once, using a single password, and get authenticated access to all network resources—without sending any passwords over the network.

slot The portion of a PKCS #11 module that contains a security device. A slot can be implemented in either hardware or software.

smart card A small device (typically about the size of a credit card), that contains a microprocessor and is capable of storing keys and certificates, as well as performing cryptographic operations. Smart cards implement some or all of the PKCS #11 interface.

SNMP See **Simple Network Management Protocol (SNMP)**.

SNMP master agent Software that exchanges information between SNMP subagents and a network management station.

SNMP subagent Software that gathers information about a managed device and passes the information to the SNMP master agent.

Socket Another term for a logical port through which communication takes place.

spoofing The act of pretending to be someone else. Examples: a person pretending to have the email address `jdoe@iplanet.com`, or a computer that identifies itself as `www.iplanet.com` when it is not. Spoofing is one form of impersonation. See also **misrepresentation**, **impersonation**.

SSL See **Secure Sockets Layer (SSL)**.

SSL handshake An exchange of messages that allows the server to authenticate itself to the client using public-key techniques, and then allows the client and the server to cooperate in the creation of symmetric keys.

SSL session The period of interaction between a server and a client that follows the SSL handshake.

static group A group that only changes when an administrator adds or removes members.

subagent See **SNMP subagent**.

subject The person, company, or other entity identified by the subject name of a certificate.

subject name A distinguished name (DN) that uniquely describes the person, company, or other entity that a certificate is issued for.

symmetric key encryption An encryption method that uses the same cryptographic key to encrypt and decrypt a given message.

symmetric keys A pair of keys that are used for rapid encryption, decryption, and tamper detection during an SSL session.

TCP/IP Transmission Control Protocol/Internet Protocol. The main network protocol for the Internet and for enterprise (company) networks.

token See **security device**.

topology A hierarchical representation of all the resources that are registered in a configuration directory.

trap message Messages sent by a managed device to the network management station.

trust database A collection of trusted certificates and public keys.

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

URL Uniform Resource Locator. The addressing system used by servers and clients when requesting documents. A URL is often called a location. The format of a URL is [protocol]://[machine:port]/[document]. The port number is necessary only on selected servers, and it is often assigned by the server.

Sample URLs: `http://www.iplanet.com/index.html`

`ldap://directory.iplanet.com:4345/o=iplanet.com`

user directory Typically, a directory subtree containing user and group entries. In large deployments, the user directory can be a separate instance of Directory Server.

A

access control

- examples of 172
- overview 171–174
- to navigation tree 173

Access Control Instruction, *See* ACI

Access Control List, defined 171

access log

- defined 116
- viewing in Console 117
- viewing with Administration Express 65

access permission, *See* ACI

access settings, for Administration Server 123–124

ACI

- bind rules 175
- creating with ACI Editor 177–182
- defined 171
- editing with ACI Editor 182
- removing 182
- See also* ACI Editor
- See also* ACI Manager
- See also* ACL

ACI Editor

- creating a new ACI with 177–182
- described 174
- editing an existing ACI with 182

ACI Manager

- described 174
- opening 176

ACL, defined 171

add, command for `modutil` 148

`admconfig`

options 136

overview and syntax 135

tasks 137

usage examples 144

Admin Server, *See* Administration Server

`admin_ip.pl`, overview and usage 145

administration domain

- changing user directory settings for 130
- creating 52
- creating and modifying 52–53
- defined 51
- modifying 53
- removing 54

Administration Express

- accessing 66
- overview and usage 65–68
- setting the refresh rate for 68–69
- starting or stopping server instances with 67
- viewing access and error logs in 68
- viewing basic server information in 68

administration page, using 118

Administration Server

- access settings for 123–124
- changing IP address for, *See* `admin_ip.pl`
- configuring from the command line, *See* `admconfig`
- defined 22
- directory settings for 127–134
- encryption settings for 125
- installation of 26
- instances of 77
- logging options for 116
- network settings for 121–123

- setting paths for log files 118
 - using SSL on 125
 - starting and restarting 113–115
 - stopping 115
 - storage of URLs by iPlanet Console 46
- Administration Server Administrator
 - changing user name or password for 109
 - defined 95
- administrators, overview of 95
- algorithm, cryptographic 239
- alias
 - directory containing certificate information 193
 - nickname for organizational unit 106
- appearance, customizing Console's 58
- attributes
 - defined 86
 - syntax 87
- authentication
 - certificate-based 248–250
 - client and server 246
 - used in form signing 253
 - during login to Console 130
 - password-based 246–247
 - See also* client authentication
 - See also* server authentication

B

bind rules, *See* ACI

C

- c, RDN keyword 85
- CA
 - certificate 251
 - defined 245
 - hierarchies and root 259
 - trusted 258
 - trusted CA certificate 187
- Certificate Authority Workstation, defined 233
- Certificate Authority, *See* CA.

- certificate database
 - backing up 193
 - restoring from a backup 193
- certificate group
 - creating 104
 - defined 99
- certificate request, sending as email 190
- Certificate Revocation List, *See* CRL
- certificate-based authentication, defined 246
- certificates
 - authentication using 248
 - backing up 191
 - CA certificate 251
 - certificate database 186
 - chains 260
 - checking expiration date of 196
 - client 202–209
 - contents of 255
 - generating renewal request for 196–198
 - installing 191
 - issuing of 265
 - and LDAP Directory 266
 - use during login 46–49
 - object-signing 251
 - overview of renewal 267
 - revoking 267
 - S/MIME 251
 - self-signed 259
 - server certificate 187
 - verifying a certificate chain 261
- certmap.conf
 - defined 204
 - editing 207
 - examples 208
 - See also* client authentication
- changepw, command for modutil 149
- cipher suites, defined 184
- ciphers
 - choosing 184
 - defined 239
 - option for modutil 151
 - overview 184–185
 - preferences 195
- CKL
 - obtaining and using 201–202
- client authentication

- client SSL certificates defined 250
- enabling on Administration Server 195
- logging in to iPlanet Console using 46–49
- overview of 202–203
- preparing to use 203
- setting up between servers 210
- using `certmap.conf` 204–207
- Client Authentication for Users 211
- cloning, defined 78
- `CmapLdapAttr`, `certmap.conf` property 206
- `cn`, RDN keyword 85
- community string
 - adding with iPlanet Console 223–225
 - defined 222
- Compromised Key List, *See* CKL
- `continueOnError`, option for `admconfig` 136
- Configuration Administrator
 - changing user name or password for 107
 - defined 95
- Configuration Administrators group
 - adding users to 101–102
 - defined 99
- configuration directory
 - changing settings for 128
 - defined 21
 - merging two 79–82
 - overview 127
 - See also* Directory Server
- connection restrictions, defined 121
- Console, *See* iPlanet Console
- `countAccessLogEntries`, `admconfig` task 137
- `countErrorLogEntries`, `admconfig` task 138
- `create`, command for `modutil` 149
- CRL
 - defined 235
 - managing 200
- crypto cards
 - certification process 234
 - used by Fortezza 233
- custom views
 - creating 61
 - overview 54
 - using 63–65
- customization, *See* preferences

D

- `dbdir`, option for `modutil` 151
- `dc`, RDN keyword 85
- `default`, command for `modutil` 149
- `delete`, command for `modutil` 149
- digital signatures
 - defined 242
 - use of during SSL authentication 184
- directory
 - changing the search directory 90
- directory entries
 - creating 91–106
 - removing 109
 - searching for 89
- Directory Server
 - attributes 86
 - authenticating against 171
 - common attributes in 86
 - configuration subtree 21
 - DN and attribute syntax 87
 - failover support 130
 - installing 26
 - LDAP URL 103
 - mapping client certificate to 202–209
 - merging two configuration directories 79–82
 - role in managing resources and users 21
 - user subtree 21
 - See also* configuration directory
 - See also* user directory
- `disable`, command for `modutil` 149
- `disableD[SGWAccess]`, `admconfig` task 138
- display fonts, *See* fonts
- display preferences, *See* preferences
- distinguished name, *See* DN
- DN
 - defined 84
 - overview 83
 - syntax 87
- `DNComps`, `certmap.conf` property 205
- dynamic group
 - creating 102
 - defined 99

E

- email, signed and encrypted 252
- enable, command for `modutil` 149
- enableD[SGWAccess], `admconfig` task 138
- enc[ryption], option for `admconfig` 136
- encryption
 - defined 239
 - overview of SSL 183
 - public-key 240
 - settings for Administration Server 125
 - symmetric-key 239
 - using external devices 185
- entries, *See* directory entries
- error log
 - defined 116
 - viewing in Console 117–118
 - viewing with Administration Express 65
- external security device
 - defined 186
 - See also* security device
- external token, *See* security device

F

- failover
 - user directory support for 130
- FilterComps, `certmap.conf` property 205
- fips, command for `modutil` 150
- font profiles, *See* fonts
- fonts
 - setting display 55–57
- force, command for `modutil` 150
- form signing, defined 253
- Fortezza
 - crypto cards 233
 - defined 233
 - enabling 236

G

- GET, type of SNMP message 217
- getAc[cessLog], `admconfig` task 139
- getAdd[resses], `admconfig` task 139
- getAdminUI[D], `admconfig` task 139
- getAdminUs[ers], `admconfig` task 139
- getCa[cheLifetime], `admconfig` task 140
- getCl[assname], `admconfig` task 140
- getDe[faultAcceptLanguage], `admconfig` task 140
- getDS[Config], `admconfig` task 140
- getErrorLog], `admconfig` task 143
- getH[osts], `admconfig` task 143
- getO[neACLDir], `admconfig` task 143
- getPo[rt], `admconfig` task 143
- getSe[rverAddress], `admconfig` task 144
- getSu[iteSpotUser], `admconfig` task 144
- getU[GDSConfig], `admconfig` task 141
- givenName, Directory Server attribute 86
- global keys, *See* JAR information file
- groups
 - Configuration Administrators 99
 - creating certificate group 104
 - creating dynamic group 102
 - creating static group 100
 - defined 99
 - editing 107
 - locating 89
 - removing 109

H

- h[elp], option for `admconfig` 136
- help, getting from within Console 15–17
- host information, modifying 77
- host restriction, defined 121
- HTML-based administration, using Administration Express 65–68

I

- i[inputFile], option for admconfig 136
- information panel, defined 51
- InitFn, certmap.conf property 206
- installation
 - Administration Server 26
 - Directory Server 26
 - Express Mode 27
 - modes 26–27
 - of a stand-alone Console 27–29
 - overview 25–27
 - silent 36
 - uninstalling iPlanet software 37–39
 - upgrading a stand-alone version of iPlanet Console 33–35
 - upgrading Administration Server and Console 30–35
- installdir, option for modutil 151
- instance, *See* server instance
- internal token, *See* security device
- iPlanet Console
 - defined 21
 - information panel 51
 - installing as a standalone application 27
 - logging in to ??–49
 - menus 49
 - overview of 21–24
 - storage of five Administration Server URLs 46
 - tabs 50
- iPlanet Setup Program 25

J

- JAR information file
 - global keys 157–158
 - per-file keys 160–162
 - per-platform keys 159–160
 - syntax 157–162
 - using with modutil 155–156
 - See also* modutil
- jar, command for modutil 150

K

- key-pairs, overview 186
- keys
 - defined 239
 - management and recovery 266

L

- l, RDN keyword 85
- LDAP URL, constructing 103
- ldapdelete, defined 146
- ldapmodify, defined 146
- ldapsearch, defined 146
- libfile, option for modutil 151
- library, certmap.conf property 206
- list, command for modutil 150
- Litronic cryptographic module 185
- logging in to iPlanet Console 45
- logs
 - setting new paths for 118
 - viewing access 117
 - viewing error 117

M

- mail, Directory Server attribute 86
- managed device, defined 213
- management information base, *See* MIB
- management window, opening for iPlanet server 75
- Manual ACI Editor, *See* ACI Editor
- master agent
 - configuring 222–227
 - starting from the command line 229–230
 - starting with iPlanet Console 228
- master agent, defined 214
- mechanisms, option for modutil 152
- members, adding to static group 101
- menus, in iPlanet Console 49
- MIB

- Administration Server 216
- defined 215

modutil

- commands 148–151
- options 151–153
- overview and syntax 147–148
- usage examples 162–167
- using JAR information file with 155–156
- See also* JAR information file

N

native agent

- defined 218
- reconfiguring 222
- restarting 221

navigation tree

- custom views of 61
- overview 50
- setting access permissions to 173

- network management station, defined 214

- network settings, Administration Server 121–123

- newpwfile, option for modutil 152

- nocertdb, option for modutil 153

- password-based authentication, defined 246–247

- per-file keys, *See* JAR information file

- permission, *See* ACI

- per-platform keys, *See* JAR information file

PKCS #11 module

- defined 185
- installing 186
- removing 186

- port number, defined 121

pre-4.0 server

- adding to navigation tree 71–73
- migrating to newer version 73–74

preferences

- display 54–55
- font 55–57
- overview 54
- UI permissions 54

- private key, defined 240

proxy agent

- defined 220
- installing 221
- starting 221

public key

- cryptography 238
- defined 240
- infrastructure 264
- management 266

- pwfile, option for modutil 153

O

- o, RDN keyword 86

- object signing 254

organizational units

- creating 106
- removing 109

- ou, RDN keyword 86

P

password

- changing for a user or administrator 107–109
- using for authentication 246

R

- r[estart], admconfig task 144

- RA, *See* Registration Authority

RDN

- defined 84
- keywords 85

- refresh rate, setting for Administration Express 68–69

- Registration Authority, defined 268

- relative distinguished name, *See* RDN

- renewal request, generating for certificate 196–198

- resources, defined 50

restart, Administration Server 113–115
rules, *See* ACI

S

S/MIME certificate 251

searching

- changing the search directory 90
- for directory entries 89

sec-activate, overview and syntax 146

sec-migrate, overview and syntax 146

Secure Sockets Layer, *See* SSL

security device

- defined 185–186
- installing external 186
- removing external 186

Security Module Database Tool, *See* modutil

self-signed certificate 259

ser[ver], option for admconfig 136

server

- adding a pre-4.0 72
- cloning 78
- defined 51
- installing a new 26
- opening a management window for 75
- requesting a certificate for 188–190
- starting and stopping with Administration Express 65

server certificate chain, defined 187

server certificate request, generating 188–190

server group

- changing user directory settings for 132
- defined 51
- modifying information for 77

server instance

- Administration Server 26
- creating 76
- modifying information for 77
- removing 78
- See also* server

server management window, *See* management window

server, pre-4.0

- migrating from 73

SET, type of SNMP message 217

Set Permissions dialog box, described 174

setAc[cessLog], admconfig task 139

setAdd[resses], admconfig task 139

setAdminP[wd], admconfig task 139

setAdminUI[D], admconfig task 139

setAdminUs[ers], admconfig task 140

setCa[cheLifetime], admconfig task 140

setCl[assname], admconfig task 140

setDe[faultAcceptLanguage], admconfig task 140

setDS[Config], admconfig task 141

setE[rrorLog], admconfig task 143

set[Hosts], admconfig task 143

setO[neACLDir], admconfig task 143

setPo[rt], admconfig task 144

setSe[rverAddress], admconfig task 144

setSu[iteSpotUser], admconfig task 144

setU[GDSConfig], admconfig task 142

Setup Program 25

silent installation 36

single sign-on 253

slot, defined 185–186

slot, option for modutil 153

SMUX, defined 215

sn, RDN keyword 86

SNMP

- community string 222

- examples of message transfer 217

- installing a proxy agent 221

- managed devices defined 213

- enabling master agent 228

- master agent defined 214

- messages defined 217

- MIB defined 215

- multiplexing protocol (SMUX) defined 215

- native agent defined 218

- network management station defined 214

- overview 213–216

- proxy agent 220

- proxy agent defined 220

- setting up 218–219

- setting up on Windows NT 230

- starting a proxy agent 221
- enabling subagent 223
- subagent defined 214
- trap destinations 223
- See also* master agent
- See also* subagent
- SNMP master agent, *See* master agent
- SNMP native agent, *See* native agent
- SNMP proxy agent, *See* proxy agent
- SNMP subagent, *See* subagent
- SSL
 - activating on Netscape and iPlanet servers 194
 - using with Administration Server 125
 - backward compatibility of 184
 - ciphers, *See* ciphers
 - client 202–209
 - client certificates 250
 - defined 183
 - editing `certmap.conf` 207
 - examples of `certmap.conf` 208
 - external security device 186
 - generating a certificate request 188–190
 - internal security device 186
 - overview of protocol 183–188
 - preparing to set up 188
 - sending a manual certificate request 190
 - slots and security devices 185
- st, RDN keyword 86
- st[op], `admconfig` task 144
- stand-alone Console, installation 27–29
- static group
 - creating 100
 - defined 99
- streetAddress, Directory Server attribute 87
- subagent
 - defined 214
 - See also* SNMP
- synchronization options
 - enabling 96
 - overview 96
 - setting 97–98
- sysContact, defining in master agent CONFIG file 227
- sysLocation, defining in master agent CONFIG file 227

T

- tables
 - changing column position in 59
 - changing width of columns in 60
 - customizing 59–61
- tabs, in iPlanet Console 50
- target, *See* ACI
- TCP/IP, defined 237
- telephoneNumber, Directory Server attribute 87
- tempdir, option for `modutil` 153
- title, Directory Server attribute 87
- TLS
 - defined 183
 - See also* SSL
- To Set Up Client Authentication for Users 211
- token, *See* security device
- topology
 - defined 50
 - See also* navigation tree
- Transport Layer Security, *See* TLS
- trap messages, defined
- traps
 - adding destination with iPlanet Console 225
 - See also* SNMP
 - See also* trap messages
- trusted CA, defined 258

U

- u[ser], option for `admconfig` 137
- uid, Directory Server attribute 87
- undefault, command for `modutil` 150
- uninstallation 37–39
- upgrade
 - of a stand-alone version of iPlanet Console 33–35
 - See also* installation
- user authentication, *See* authentication
- user directory
 - failover support 130
 - overview 129
 - settings 129

See also Directory Server

- user entries
 - administrators 95
 - changing passwords for 107
 - creating 91
 - editing 107
 - locating 89
 - preferred language of 94
 - removing 109
- userPassword, Directory Server attribute 87
- Users and Groups tab, changing the search directory for 90

V

- verb[ose], option for admconfig 137
- verifycert, certmap.conf property 206
- vers[ion], option for admconfig 137
- view, *See* custom views
- viewA[ccessLogEntries], admconfig task 138
- viewE[rrorLogEntries], admconfig task 138
- Visual ACI Editor, *See* ACI Editor

