Administrator's Guide

Netscape Directory Server

Version 4.1


Recycled and Recyclable Paper

Version 4.1

01 00 99 10 9 8 7 6 5 4 3 2 1

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

Contents

Introduction

Welcome to Netscape Directory Server and the Internet. Netscape Communications Corporation is the premier provider of open software that lets people and companies exchange information and conduct commerce over enterprise networks and the Internet.

This *Administrator's Guide* documents the Netscape Directory Server and the Restricted Mode Directory Server, products that supply management and retrieval of corporate information.

Netscape Directory Server Restricted Mode

A version of the Netscape Directory Server is bundled with some systems. This server provides a restricted subset of features. You can purchase an upgrade to the full version of the server from Netscape. See "Netscape Directory Server 4.1 Overview" on page 22 for information on the full-featured directory. The restricted directory server provides the following functionality:

- Netscape Console—A powerful server management tool that uses a graphical interface. You can log in from any system connected to your network to manage a remote server or to make changes in a centralized directory.
- Command-line tools—Allow you to script updates and other modifications to your directory server and its contents.
- Schema management interface—Lets you create custom object classes and attributes to define entries specific to your enterprise's needs.
- Import and export LDIF files—Help you manage directory entries and allows you to add, modify, and delete multiple entries.

- Backup and restore database—Allow you to make backups of the directory database and restore from the backups to protect against data loss.
- SSL—Provides secure communications over the network including ciphers with up to 40-bit encryption.

Netscape Directory Server 4.1 Overview

In addition to the functionality delivered with the restricted mode of the Directory Server, the full version of Netscape Directory Server 4.0 provides the following:

- FORTEZZA—An encryption system used by federal and government agencies to manage sensitive but unclassified information.
- Directory Server Gateway—Customizable HTTP to LDAP client that allows you to access directory data from a web browser.
- Netscape Directory Express—Basic directory lookup tool that you can use right out of the box.
- Replication and referrals—Let you extend your directory service beyond a single server configuration.
- Support for SNMP—Permits you to monitor your directory server in real time using the Simple Network Management Protocol (SNMP).
- NT Synchronization Service—Allows you to synchronize the entries in your Windows NT directory with the entries in your Netscape Directory Server directory.
- Password Policy and Account Lockout—Allows you to define a set of rules that govern how passwords and accounts are managed in the directory server.
- Plug-in interface—Allows you to replace the back-end data store, set up triggers that provide data validation and notification, customize the authentication process, and extend the functionality of the server through custom plug-ins.

- Online backup and restore—Allows you to create backups and restore from backups while the server is running.
- Additional SSL encryption cipher support; up to 168 bit.

Prerequisite Reading

This manual describes how to administer the directory server and its contents. This manual also describes how to administer the NT Synchronization Service. However, this manual does not describe many of the basic directory and architectural concepts that you need to successfully deploy, install, and administer your directory service. Those concepts are contained in the *Netscape Directory Server Deployment Manual*. You should read that book before continuing with this manual.

After you are familiar with directory server concepts and have done some preliminary planning for your directory service, you can install the Netscape Directory Server. The instructions for installing the various Directory Server components are contained in the *Netscape Directory Server Installation Guide*.

Also, *Managing Servers with Netscape Console* contains general background information on how to use Netscape servers. You should read and understand the concepts in that book before you attempt to administer the Netscape Directory Server.

What Is in This Book?

This manual explains how to administer the Netscape Directory Server and the NT Synchronization Service. Before you read this book, you should read the *Netscape Directory Server Deployment Manual*. That manual documents server concepts.

After configuring your server, use this manual to help maintain your server.

Conventions Used in This Book

This section explains the conventions used in this book.

`Monospaced font`—This typeface is used for any text that appears on the computer screen or text that you should type. It is also used for filenames, functions, and examples.

Note Notes and Warnings mark important information. Make sure you read the information before continuing with a task.

|—The vertical bar is used as a separator for user interface elements. For example, Configuration|Logs means you should go to the Configuration tab on the Directory Server Console and then select the Logs icon.

Throughout this book you will see path references of the form

```
<NSHOME>/slapd-<serverID>/...
```

In these situations, `<NSHOME>` represents the directory where you installed the server, and `<serverID>` represents the server identifier you gave the server when you installed it. For example, if you installed your server in `/export/ns-home` and gave the server an identifier of `phonebook`, then the actual path would be

```
/export/ns-home/slapd-phonebook/. . .
```

Also, all paths specified in this manual are in Unix format. If you are using a Windows NT-based directory server, you should assume the NT equivalent file paths whenever Unix file paths are shown in this book.

Chapter

1

Administering Netscape Directory Server

The Netscape Directory Server simplifies management and retrieval of corporate user information. Using the directory server, corporate IS organizations can manage all their user information from a single point of control, and corporate users can retrieve this information from multiple, easily accessible network locations.

The Netscape Directory Server product ships with a directory server, an administration server, and Netscape Console.

This chapter provides the information you need to get started administering the directory server, in the following sections:

Overview of Directory Server Management

Netscape Directory Server is based on an open-systems server protocol called the Lightweight Directory Access Protocol (LDAP). The directory server is a robust, scalable server designed to manage an enterprise-wide directory of users and resources. The directory server runs as the `ns-slapd` process or service on your machine. The server manages the directory databases and responds to client requests.

You perform most Directory Server administrative tasks through the Administration Server, a second server that Netscape provides to help you manage the Directory Server (and all other Netscape Servers). For Directory Server, you use a part of the Administration Server called Netscape Console. The *Directory Server Console* is a part of Netscape Console designed specifically for use with Netscape Directory Server.

You can perform most directory server administrative tasks from the Directory Server Console. You can also perform administrative tasks manually by editing the configuration files or by using command-line utilities. For more information about the Netscape Console see *Managing Servers with Netscape Console.*

Using the Directory Server Console

From the Directory Server Console you can do the following:

- Start and stop the directory server
- Backup and restore directory databases
- Manage access control, referrals, directory schema, and server settings such as TCP ports and the database cache size
- Manage the directory server configuration files and directory content
- View and configure server logs

Opening the Directory Server Console

You bring up the Directory Server Console from Netscape Console, which is described in *Managing Servers with Netscape Console*. See *Installing the Netscape Directory Server* for information on installing the server.

To open the Directory Server Console, from the Netscape Console:

1. On the Console tab, open the folder designated by the domain in which the directory server resides, for example, `airius.com`.
2. Open the folder designated by the hostname of the directory server, for example, `dirserver.airius.com`.
3. Expand the Server Group folder.
4. Double-click the Directory Server entry (for example, `slapd-phonebook`).

   This brings up the Directory Server Console with the Tasks tab displayed by default.

Binding to the Directory From Netscape Console

When you create or manage entries from the Directory Server Console, and when you first access the Netscape Console, you are given the option to log in by providing a bind DN and a password. This option allows you to indicate who you are accessing the directory tree as. This in turn determines whether you can perform the requested operation in the tree.

You can log in with the root DN when you first bring up the Netscape Console. If you choose not to do this, you can log in as the root DN or a different user through the Directory Server Console.

To log in to Netscape Console:

1. On the Directory Server Console, select the Tasks tab.
2. Click “Log on to the Directory Server as a New User”.

   A login dialog box appears.
3. Enter the new DN and password and click OK.

   Enter the full distinguished name of the entry with which you want to bind to the server. For example, if you want to bind as the Root DN and the Root DN is Directory Manager, then enter the following in the Distinguished Name text box:

   ```
   cn=Directory Manager
   ```

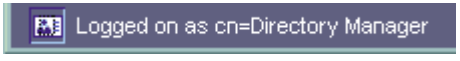
   For more information about the root DN and password, refer to “Managing the Root DN” on page 288.

   Do not perform daily administrative tasks using the directory manager as your bind DN. Instead, set up a directory server administrator account with the access control privileges required for the most common tasks you perform. For information on how to do this, see *Managing Servers with Netscape Console*.

Viewing the Current Bind DN From Netscape Console

You can view the bind DN you used to log in to the Directory Server Console by clicking the login icon in the lower-left corner of the display. The current bind DN appears next to the login icon as shown here.

Figure 1.1 Viewing the bind DN



Starting and Stopping the Directory Server

If you are not using Secure Sockets Layer (SSL), you can start and stop the directory server using the methods listed here. If you are using SSL, see "Starting the Server with SSL Enabled" on page 30.

From the Directory Server Console. On the Tasks tab, click "Start the Directory Server" or "Stop the Directory Server" as appropriate.

When you successfully start or stop your directory server from the server console, the server displays a message box stating either that the server started or has shut down.

From the Windows NT Services Control Panel.

1. Select Start|Settings|Control Panel from the desktop.
2. Double-click the Services icon.
3. Scroll through the list of services and select the Netscape Directory Server.

   The service name is `Netscape Directory Server 4.1 (<serverID>)` where *<serverID>* is the identifier you gave the server when you installed it.
4. Start or stop the service:

   - To stop the service, click Stop and then confirm that you want to stop the service.
   - To start the service, select the directory server service and click Start.

From the Unix or Windows NT command line. Use one of the following scripts:

`<NSHOME>/slapd-<serverID>/start-slapd`

or

`<NSHOME>/slapd-<serverID>/stop-slapd`

where *<NSHOME>* is the location where your server is installed, and *<serverID>* is the identifier you gave the server when you installed it.

On Unix, both of these scripts must run with the same UID and GID as that used by the directory server. For example, if the directory server runs as `nobody`, you must run the `start-slapd` and `stop-slapd` utilities as `nobody`.

Starting the Server with SSL Enabled

On Windows NT, if you are using SSL with your server, then you must start the server from the server's host machine. This is because a dialog box will prompt you for the certificate PIN before the server will start. For security reasons, this dialog box appears only on the server's host machine.

On Unix, you must start the server from the command line.

Note If you are using FORTEZZA, see "Starting the Server with FORTEZZA Enabled" on page 322 for information on starting and stopping the server.

Alternatively, on either platform, you can create a password file to store your certificate password. By placing your certificate database password in a file, you can start your server from the server console, and also allow your server to automatically restart when running unattended.

This password is stored in clear text within the password file, so its usage represents a significant security risk. Do not use a password file if your server is running in an unsecured environment.

The password file must be placed in the following location:

`<NSHOME>/alias/slapd-<serverID>-pin.txt`

where `<NSHOME>` is the location where your server is installed, and `<serverID>` is the identifier you gave the server when you installed it.

You create certificate databases using the administration server and the Certificate Setup Wizard. For information on certificate databases, certificate aliases, SSL, and obtaining a server certificate, see *Managing Servers with Netscape Console*. For information on using SSL with your directory server, see Chapter 11, "Managing SSL."

Starting the Server in Referral-Only Mode

You can also start the server in referral-only mode. You might want to do this if you're making configuration changes to the directory server and you want all clients to be referred to another master for the duration. There are two ways to configure the server to start up in referral-only mode:

- Add the following directive to `slapd.conf` and restart the server:

  ```
  referralmode <url>
  ```

  where `<url>` is the LDAP URL you want the server to send to clients.

- Start the server with the refer command as follows:

  On Unix—Change to the directory:

  ```
  <NSHOME>/bin/slapd/server/
  ```

  and then run the refer command as follows:

  ```
  ./ns-slapd refer -p <port> -r <url>
  ```

  On Windows NT—run the refer command as follows:

  ```
  <NSHOME>\bin\slapd\server\slapd refer -p <port>
  -r <url>
  ```

  where `<NSHOME>` is the directory where you installed the directory server, `<port>` is the port number of the directory server you want to start in referral-only mode, and `<url>` is the LDAP URL you want to return to clients. If you use this option, the server starts up without reading `slapd.conf`. You should use this option if the database is undergoing maintenance or is otherwise unavailable.

Using the Command-Line Utilities

Netscape Directory Server comes with a robust set of command-line utilities that you can use to manage the entries in your directory. The most important of these are listed in Table 1.1.

Table 1.1 Commonly used command-line utilities

Command-line utility	Description
`aclupg`	Upgrades LDIF formatted with the 1.x access control statements to the 4.x ACI. See the *Netscape Directory Server Installation Guide* for more information.
`ldapdelete`	Allows you to delete entries in the directory. For information on using this utility, see "Deleting Entries Using ldapdelete" on page 247.
`ldapsearch`	Allows you to search the directory. Returns search results in LDIF format. For details on this tool, see Chapter 8, "Finding Directory Entries."
`ldapmodify`	Allows you to add, delete, modify, or rename entries. All operations are specified using LDIF update statements. For details on this tool, see "Adding and Modifying Entries Using ldapmodify" on page 243.
`ns-slapd` (Unix) `slapd` (Windows NT)	Used to start the directory server process, to build a directory database from an LDIF file, or to convert an existing database to an LDIF file. For details, see • "Starting and Stopping the Directory Server" on page 29 • "Importing LDIF From the Command Line" on page 77 • "Exporting to LDIF From the Command Line" on page 71
`ldif`	Automatically formats LDIF files for you, and creates base 64 encoded attribute values. For details on this tool, see "Base 64 Encoding" on page 44.

Finding the Command-Line Utilities

Most of the directory server's command line utilities are stored in a single location. You can find them in the following directory:

`<NSHOME>/bin/slapd/server`

where <*NSHOME*> is your server installation directory.

The remaining three—`ldapdelete`, `ldapmodify`, and `ldapsearch`—are stored in the following directory:

`<NSHOME>/shared/bin`

where <*NSHOME*> is your server installation directory.

Warning The command-line utilities in these directories that are not described in this manual are used internally by the directory server. Their use outside of that environment is not recommended.

Setting Environment Variables

On Windows NT, before using the command-line utilities, set your PATH variable to include the locations of the directory server command-line utilities:

`<NSHOME>/bin/slapd/server`

and

<NSHOME>/shared/bin

For information on how to set environment variables, see the documentation available for your operating system.

On Unix, to run the command-line utilities, change to the directory where they are stored.

Directory Server Command-Line Scripts

In addition to the command-line utilities described in “Using the Command-Line Utilities” on page 32, the Netscape Directory Server provides several scripts you can use to invoke the utilities with the most common options set. These scripts are stored in the following directory:

`<NSHOME>/slapd-<serverID>/`

All of these scripts assume that you want to use the `slapd.conf` file located in

`<NSHOME>/slapd-<serverID>/config/`

You can copy these scripts and modify your copies to suit your needs. In general, the rest of this manual does not describe the use of these scripts. Some of the most commonly used scripts are listed in Table 1.2.

Table 1.2 Commonly used command-line scripts

Command-line script	Description
`bak2db`	Restores the database from the most recent archived backup. Syntax: `bak2db [backup_directory]`
`db2bak`	Creates a backup of the current database contents. Syntax: `db2bak [backup_directory]`. For more information, see “Backing Up Your Database From the Command Line” on page 82.
`db2ldif`	Exports the contents of the database to LDIF. By default, the server stores the LDIF file in: `<NSHOME>/slapd-<serverID>/ldif/` Syntax: `db2ldif <ldif_filename> [-s <include suffix>] [-x <exclude suffix>]`
`getpwenc`	Prints the encrypted form of a password using one of the server's encryption algorithms. If a user cannot log in, you can use this script to compare the user's password to the password stored in the directory. Syntax: `getpwenc sha <password>` or: `getpwenc crypt <password>`

Table 1.2 Commonly used command-line scripts (Continued)

Command-line script	Description
`ldif2db`	Runs the `slapd` (Windows NT) or `ns-slapd` (Unix) command-line utility with the `ldif2db` keyword. By default, the script first saves and then merges any existing configuration tree (`o=NetscapeRoot`), with any files to be imported. You can specify `-noconfig` if you want to overwrite the configuration information. **Warning.** Netscape recommends that you do not overwrite the configuration data unless instructed to do so by Netscape Techical Support. Syntax: `ldif2db [-noconfig] -i <ldif filename> [-i <ldif filename>] ... [-s <include suffix>] [-x <exclude suffix>}`
`monitor`	Retrieves performance monitoring information using the `ldapsearch` command-line utility. Syntax: `monitor -b <baseDN> [options] filter` or: `monitor "cn=monitor" <port>` See “Using ldapsearch” on page 212 for more information on `ldapsearch`.
`restart-slapd`	Restarts the directory server. Syntax: `restart-slapd`
`start-slapd`	Starts the directory server. Syntax: `start-slapd`
`stop-slapd`	Stops the directory server. Syntax: `stop-slapd`
`vlvindex`	Reserved.

Directory Server Configuration Files

You can also perform many administrative tasks manually by editing the directory server's configuration files. There are two main configuration files:

- `slapd.conf`—A text (UTF-8) file that contains the server's configuration and parameter values. These parameters are read by the server only at startup time, and define all of the server parameters that are not related to the server's database, for example, the server's name, the port that it uses, and performance tuning values. The `slapd.conf` file and its parameters are described in detail in Chapter 17, "Configuration Parameters."
- `slapd.ldbm.conf`—Included in the `slapd.conf` using the `dynamicconf` parameter. This file contains the directory server's database parameters.

All of the directory server's configuration files are located in the following directory:

`<NSHOME>/slapd-<serverID>/config`

where <*NSHOME*> is your server installation directory and <*serverID*> is the server identifier that you defined when you installed your directory server. Thus, if you installed your directory server in `/usr/dirserver` and you selected a server identifier of `phonebook`, then your configuration files are all stored under

```
/usr/dirserver/slapd-phonebook/config
```

Table 1.3 briefly describes each configuration file.

Table 1.3 Directory Server Configuration Files

Configuration Filename	Purpose
dse.ldif	Contains front-end Directory Specific Entries created by the directory at server startup. These include the Root DSE (""), and the contents of `cn=config`, `cn=monitor`, and `cn=schema`.
ldbm.ldif	Contains back-end Directory Specific Entries created by the directory at server startup. These include the contents of `cn=config,cn=ldbm` and `cn=monitor,cn=ldbm`.
ns-admin-schema.conf	Schema used by Netscape Administration Server 4.0 and Netscape Console.
ns-calendar-schema.conf	Schema used by Netscape Calendar Server.
ns-certificate-schema.conf	Schema used to identify a Netscape Certificate Server. netscapeCertificateServer is the sole object class.
ns-common-schema.conf	Schema that contains objects classes and attributes common to the Netscape Console framework.
ns-compass-schema.conf	Schema used by Netscape Compass Server to define personal interest profiles.
ns-delegated-admin-schema.conf	Schema used by Netscape Delegated Administrator 1.0.
ns-directory-schema.conf	Schema used to identify a Netscape Directory Server.
ns-legacy-schema.conf	Schema used by Netscape Administration Server for legacy servers.
ns-mail-schema.conf	Schema used by Messaging Server to define mail users and mail groups.
ns-mcd-browser-schema.conf	Schema used by Mission Control Desktop to hold browser client preferences.
ns-mcd-config-schema.conf	Schema used by Mission Control Desktop to set MCD "config()" preferences.

Table 1.3 Directory Server Configuration Files (Continued)

Configuration Filename	Purpose
ns-mcd-li-schema.conf	Schema used by Mission Control Desktop to define location independence.
ns-mcd-mail-schema.conf	Schema used by Mission Control Desktop to hold mail client preferences and messenger security preferences.
ns-media-schema.conf	Schema used to identify a Netscape Media server.
ns-mlm-schema.conf	Schema used by Messaging Server 4.0 for mailing list management.
ns-msg-schema.conf	Schema used by Netscape Messaging Server 4.0.
ns-netshare-schema.conf	Schema used by Netscape Enterprise and FastTrack servers.
ns-news-schema.conf	Schema used by Netscape Collabra Server to hold news group preferences.
ns-proxy-schema.conf	Schema used to identify a proxy server.
ns-schema.conf	Lists all schema files used by the Netscape Directory Server.
ns-value-schema.conf	Schema used for defining schemaless configuration for LDAP.
ns-web-schema.conf	Schema used to identify an HTTP server.
slapd.at.conf	Contains Includes X.500 user schema for use with LDAP, LDAP attributes defined by the IETF, pilot X.500 schema for use in LDAPv3, and Netscape-defined attributes. Modifying this file will cause interoperability problems. User defined attributes should be added using Netscape Console.
slapd.conf	Contains server configuration parameters.
slapd.conf.old	Backup of slapd.conf.
slapd.ldbm.conf	Contains database configuration parameters.

Table 1.3 Directory Server Configuration Files (Continued)

Configuration Filename	Purpose
slapd.oc.conf	Contains standard object classes expected to be present in Directory Server 4.x unchanged. Modifying this file will cause interoperability problems. User defined object classes should be added using Netscape Console. User-defined objectClasses are saved in slapd.user_oc.conf.
slapd.user_at.conf	Contains user-defined attributes.
slapd.user_oc.conf	Contains user-defined object classes.
slapd-collations.conf	Contains collation orders used with matching rules.

Chapter

2

LDAP Data Interchange Format

The directory server uses the LDAP Data Interchange Format (LDIF) to describe a directory and directory entries in text format. LDIF is commonly used to initially build a directory database or to add large numbers of entries to the directory all at once. In addition, LDIF is also used to describe changes to directory entries. For this reason, most of the directory server's command-line utilities rely on LDIF for either input or output.

Because LDIF is a text file format, you can create your LDIF files using virtually any language. All directory data is stored using the UTF-8 encoding of Unicode. Therefore, the LDIF files you create must also be UTF-8 encoded.

This chapter provides information about LDIF in the following sections:

For information on using LDIF to modify directory entries, see Chapter 9, "Managing Directory Entries."

LDIF File Format

LDIF consists of one or more directory entries separated by a blank line. Each LDIF entry consists of an optional entry ID, a required distinguished name, one or more object classes, and multiple attribute definitions.

The basic form of a directory entry represented in LDIF is

```
dn: <distinguished name>
objectClass: <object class>
objectClass: <object class>
...
<attribute type>[;subtype]:<attribute value>
<attribute type>[;subtype]:<attribute value>
...
```

You must supply the DN and at least one object class definition. In addition, you must include any attributes required by the object classes that you define for the entry. All other attributes and object classes are optional. You can specify object classes and attributes in any order. The space after the colon is also optional. For information on standard object classes and attributes, refer to the *Netscape Directory Server Schema Reference Guide*.

Table 2.1 describes the LDIF fields shown in the previous definition.

Table 2.1 LDIF fields

Field	Definition
[<id>]	Optional positive decimal number representing the entry ID. The database creation tools generate this ID for you. Never add or edit this value yourself.
dn: <distinguished name>	Specifies the distinguished name for the entry. For a complete description of distinguished names, refer to the *Netscape Directory Server Deployment Manual*.
objectClass: <object class>	Specifies an object class to use with this entry. The object class identifies the types of attributes, or *schema*, allowed and required for the entry. See the *Netscape Directory Server Schema Reference Guide* for a list of standard object classes, and Chapter 3, "Extending the Directory Schema," for information on customizing the schema.

Table 2.1 LDIF fields (Continued)

Field	Definition
<attribute type>	Specifies a descriptive attribute to use with the entry. The attribute should be defined either in `slapd.at.conf` or with the `attribute` parameter in `slapd.conf`. See the *Netscape Directory Server Schema Reference Guide* for a list of standard attributes, and Chapter 3, "Extending the Directory Schema," for information on customizing the schema.
[subtype]	Optional. Specifies a subtype, either language, binary, or pronunciation. Use this tag to identify the language in which the corresponding attribute value is expressed, or whether the attribute value is binary or a pronunciation of an attribute value. For a complete list of the supported subtypes tags, see Table B.2 on page 497.
<attribute value>	Specifies the attribute value to be used with the attribute type.

The LDIF syntax for representing a change to an entry in the directory is different from the syntax described above. For information on using LDIF to modify directory entries, see Chapter 9, "Managing Directory Entries."

Continued Lines

When you specify LDIF, you can break and continue, or fold, a line by indenting the continued portion of the line by exactly one space. For example, the following two statements are identical:

```
dn: cn=Jake Lupinski, o=airius.com
```

```
dn: cn=Jake Lup
 inski, o=air
 ius.com
```

You are not required to break and continue LDIF lines. However, doing so may improve the readability of your LDIF file.

Base 64 Encoding

You can represent Binary data, such as a JPEG image, in LDIF by using base 64 encoding. You identify base 64 encoded data by using the :: symbol. For example:

```
jpegPhoto:: <encoded data>
```

In addition to binary data, other values that must be base 64 encoded include

- any value that begins with a semicolon (;) or a space
- any value that contains non-ASCII data, including newlines

Use the `ldif` command-line utility with the `-b` parameter to convert binary data to LDIF format:

```
ldif -b <attribute_name>
```

where *<attribute_name>* is the name of the attribute to which you are supplying the binary data. The binary data is read from standard input and the results are written to standard output. Thus, you should use redirection operators to select input and output files.

The `ldif` command-line utility will take any input and format it with the correct line continuation and appropriate attribute information. The `ldif` utility also senses whether the input requires base 64 encoding.

LDIF Parameter

-b Specifies that the `ldif` utility should interpret the entire input as a single binary value. If `-b` is not present, each line is considered to be a separate input value.

LDIF Example

The following example takes a binary file containing a JPEG-formatted image and converts it into LDIF format for the attribute named `jpegPhoto`. The output is saved to `out.ldif`:

```
ldif -b jpegPhoto < mark.jpg > out.ldif
```

For information on where to find command-line utilities in your directory server installation, see “Finding the Command-Line Utilities” on page 33.

Creating Directory Entries Using LDIF

There are many types of entries that you can store in your directory. This section concentrates on three of the most common types of entries used in a directory: organization, organizational unit, and organizational person entries.

The object classes defined for an entry are what indicate whether the entry represents an organization, an organizational unit, an organizational person, or some other type of entry. For a complete list of the object classes you can use by default in your directory and a list of the most commonly used attributes, see the *Netscape Directory Server Schema Reference Guide*.

Specifying Organization Entries

Directories often have at least one organization entry. Typically this is the first, or root, or topmost entry in your directory. The organization entry often corresponds to the suffix set for your directory. For example, if your directory is defined to use a suffix of `o=airius.com`, then you will probably have an organization entry in your directory named `o=airius.com`.

The LDIF that you specify to define an organization entry should appear as follows:

```
dn: <distinguished name>
objectClass: top
objectClass: organization
o: <organization name>
<list of optional attributes>
...
```

The following is a sample organization entry in LDIF format:

```
dn: o=airius.com
objectclass: top
objectclass: organization
o: Airius Corporation
description: Fictional company for example purposes
telephonenumber: 555-5555
```

The organization name in the following example uses a comma:

```
dn: o="Airius Chile\\, S.A."
objectclass: top
objectclass: organization
o: “Airius Chile\\, S.A.”
description: Fictional company for example purposes
telephonenumber: 555-5556
```

Each element of the LDIF-formatted organization entry is defined in Table 2.2.

Table 2.2 LDIF elements in organization entries

LDIF Element	Description
`dn: <distinguished name>`	Specifies the DN for the entry. DNs are described in the *Netscape Directory Server Deployment Manual*. A DN is required.
`objectClass: top`	Required. Specifies the `top` object class.
`objectClass: organization`	Specifies the `organization` object class. This line defines the entry as an organization. See the *Netscape Directory Server Schema Reference Guide* for a list of the attributes you can use with this object class.
`o: <organization name>`	Attribute that specifies the organization's name. If the organization name includes a comma, you must escape the comma by either a single backslash (on NT) or two backslashes (on Unix) and the entire organization argument must be enclosed in quotation marks. For example, to set the suffix to Airius Bolivia, S.A. you would enter `“o: Airius Bolivia\\, S.A.”` on Unix or `“o: Airius Bolivia\, S.A.”` on Windows NT.
`<list of attributes>`	Specifies the list of optional attributes that you want to maintain for the entry. See the *Netscape Directory Server Schema Reference Guide* for a list of the attributes you can use with this object class.

Specifying Organizational Unit Entries

Organizational unit entries are often used to represent major branch points, or subdirectories, in your directory tree. They correspond to major, reasonably static entities within your enterprise, such as a subtree that contains people, or a subtree that contains groups. However, the organizational unit attribute that is contained in the entry may also represent a major organization within your enterprise, such as marketing or engineering.

There is usually more than one organizational unit, or branch point, within a directory tree. For information on how to design your directory tree, see the *Netscape Directory Server Deployment Manual.*

The LDIF that you specify to define an organizational unit entry must appear as follows:

```
dn: <distinguished name>
objectClass: top
objectClass: organizationalUnit
ou: <organizational unit name>
<list of optional attributes>
...
```

The following is a sample organizational unit entry in LDIF format:

```
dn: ou=people, o=airius.com
objectclass: top
objectclass: organizationalUnit
ou: people
description: Fictional organizational unit for example purposes
```

Table 2.3 defines each element of the LDIF-formatted organizational unit entry.

Table 2.3 LDIF elements in organizational unit entries

LDIF Element	Description
`dn: <distinguished name>`	Specifies the DN for the entry. DNs are described in the *Netscape Directory Server Deployment Manual*. A DN is required. If there is a comma in the DN, the comma must be escaped with a backslash (\). For example: `dn: ou=people, o=airius bolivia\, S.A.`
`objectClass: top`	Required. Specifies the `top` object class.

Table 2.3 LDIF elements in organizational unit entries (Continued)

LDIF Element	Description
`objectClass: organizationalUnit`	Specifies the `organizationalUnit` object class. This line defines the entry as an organizationalUnit. See the *Netscape Directory Server Schema Reference Guide* for a list of the attributes you can use with this object class.
`ou: <organizational unit name>`	Attribute that specifies the organizational unit's name.
`<list of attributes>`	Specifies the list of optional attributes that you want to maintain for the entry. See the *Netscape Directory Server Schema Reference Guide* for a list of the attributes you can use with this object class.

Specifying Organizational Person Entries

The most common type of entry that you will include in your directory will probably describe a person within your organization. The majority of the entries in your directory will represent organizational people.

The LDIF you specify to define an organizational person should appear as follows:

```
dn: <distinguished name>
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: <common name>
sn: <surname>
<list of optional attributes>
...
```

The following is an example organizational person entry in LDIF format:

```
dn: uid=bjensen, ou=people, o=airius.com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Babs Jensen
sn: Jensen
givenname: Babs
uid: bjensen
ou: Marketing
ou: people
description: Fictional person for example purposes
telephonenumber: 555-5557
userpassword: {sha}dkfljlk34r2kljdsfk9
```

Table 2.4 defines each aspect of the LDIF-formatted person entry.

Table 2.4 LDIF elements in person entries

LDIF Element	Description
`dn: <distinguished name>`	Specifies the DN for the entry. DNs are described in the *Netscape Directory Server Deployment Manual*. A DN is required. If there is a comma in the DN, the comma must be escaped with a backslash (\). For example, `dn: uid=bjensen, ou=people, o=airius bolivia\, S.A.`
`objectClass: top`	Required. Specifies the `top` object class.
`objectClass: person`	Specifies the `person` object class. This object class specification should be included because many LDAP clients require it during search operations for a person or an organizational person.
`objectClass: organizationalPerson`	Specifies the `organizationalPerson` object class. This object class specification should be included because some LDAP clients require it during search operations for an organizational person.

Table 2.4 LDIF elements in person entries (Continued)

LDIF Element	Description
`objectClass: inetOrgPerson`	Specifies the `inetOrgPerson` object class. The `inetOrgPerson` object class is recommended for the creation of an organizational person entry because this object class includes the widest range of attributes. The `uid` attribute is required by this object class, and entries that contain this object class are named based on the value of the `uid` attribute. See the *Netscape Directory Server Schema Reference Guide* for a list of the attributes you can use with this object class.
`cn: <common name>`	Specifies the person's common name. That is, the full name commonly used by the person. For example, `cn: Bill Anderson`. At least one common name is required.
`sn: <surname>`	Specifies the person's surname, or last name. For example, `sn: Anderson`. A surname is required.
`<list of attributes>`	Specifies the list of optional attributes that you want to maintain for the entry. See the *Netscape Directory Server Schema Reference Guide* for a list of the attributes you can use with this object class.

Defining Directories Using LDIF

You can define the contents of an entire directory using LDIF. Use this method of directory creation when you have many entries to add to the directory.

In general, to create a directory using LDIF, follow these steps:

1. Create an ASCII file containing the entries you want to add in LDIF format.

   Make sure each entry is separated from the next by an empty line. For more information, see "Creating Directory Entries Using LDIF" on page 45.

2. Begin each directory in the database with the topmost, or root, entry.

   The root point of the directory must represent a suffix you have set for your server. For example, if your server has the suffix `o=airius.com`, then the first entry in your directory must be:

   ```
   dn: o=airius.com
   ```

   If the suffix contains a comma, the comma must be preceded by two backslashes (`\\`), to serve as escape characters, and the entire organization argument must be enclosed in quotation marks. For example, if your server has the suffix `o=Airius Bolivia, S.A.`, then the corresponding suffix entry must be

   ```
   dn: “o=airius bolivia\\, S.A.”
   ```

   For information on suffixes, see the `Suffix` parameter in Chapter 17, “Configuration Parameters.”

3. Make sure that you create an entry representing a branch point before you create new entries under that branch.

   For example, if you want to place an entry in a people and a group subtree, then create the branch point for those subtrees before creating entries within those subtrees:

   ```
   dn: o=airius.com
   <list of attributes and object classes>

   dn: ou=people, o=airius.com
   <list of attributes and object classes>

   ...

   <People subtree entries.>

   ...

   dn: ou=groups, o=airius.com
   <list of attributes and object classes>

   ...
   ```

```
<Groups subtree entries.>

...
```

If an entry's DN contains a comma, the comma must be preceded by a backslash (\). For example:

```
dn: ou=people, o=airius bolivia\, S.A.
```

4. Create the directory from the LDIF file using one of the following methods:

   - Directory Server Console—Use this method if you have a small database to import (less than 1000 entries). See "Importing LDIF From the Server Console" on page 75.
   - `ldif2db` command-line utility—Use this method if you have a large database to import (more than 1,000 entries). See "Importing LDIF From the Command Line" on page 77.
   - `ldapmodify` command-line utility with the `-a` parameter—Use this method if you currently have a directory database, but you are adding an entire new directory or subdirectory to the database. If you are adding an entire new directory to your database, make sure you define a suffix for your new directory first. Unlike the other methods for creating the directory from an LDIF file, the directory server must be running before you can add a directory or subdirectory using `ldapmodify`. See "Adding and Modifying Entries Using ldapmodify" on page 243.

LDIF File Example

The following example shows an LDIF file that contains one organization, two organizational units, and three organizational person entries:

```
dn: o=airius.com
objectclass: top
objectclass: organization

o: airius.com
description: Fictional organization for example purposes

dn: ou=People, o=airius.com
objectclass: top
objectclass: organizationalUnit
ou: People
```

```
description: Fictional organizational unit for example purposes
tel: 555-5559

dn: cn=June Rossi, ou=People, o=airius.com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: June Rossi
sn: Rossi
givenName: June
mail: rossi@airius.com
userPassword: {sha}KDIE3AL9DK
ou: Accounting
ou: people
telephoneNumber: 2616
roomNumber: 220

dn: cn=Marc Chambers, ou=People, o=airius.com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Marc Chambers
sn: Chambers
givenName: Marc
mail: chambers@airius.com
userPassword: {sha}jdl2alem87dlacz1
telephoneNumber: 2652
ou: Manufacturing
ou: People
roomNumber: 167

dn: cn=Robert Wong, ou=People, o=airius.com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Robert Wong
cn: Bob Wong
sn: Wong
givenName: Robert
givenName: Bob
mail: bwong@airius.com
userPassword: {sha}nn2msx761
telephoneNumber: 2881
roomNumber: 211
ou: Manufacturing
ou: people

dn: ou=Groups, o=airius.com
objectclass: top
```

```
objectclass: organizationalUnit
ou: groups
description: Fictional organizational unit for example purposes
```

Storing Information in Multiple Languages

If your directory contains entry and attribute information in a single language, you do not need to do anything special to add a new entry to the directory. However, if your organization is multinational, you may find it necessary to store information in multiple languages so that users in different locales can view directory information in their own language. When information in your directory is represented in multiple languages, the server associates language tags with attribute values. When you add a new entry, you must provide attribute values used in the *RDN* (Relative Distinguished Name) without any language codes.

You can even store multiple languages within a single attribute. When you do, the attribute type are the same, but each value has a different language code.

For a list of the languages supported by the directory server and their associated language tags, see “Identifying Supported Locales” on page 495.

Note The language tag has no effect on how the string is stored within the directory. All object class and attribute strings are stored using UTF-8.

For example, suppose Airius Corporation has offices in the United States and France and wants employees to be able to view directory information in their native language. When adding directory entries, the directory administrator chooses to provide attribute values in both English and French. When adding a directory entry for a new employee, Babs Jensen, the administrator creates the following LDIF entry:

```
dn: uid=bjensen, ou=people, o=airius.com
objectclass: top
objectclass: person
objectclass: organizationalPerson
name: Babs Jensen
cn: Babs Jensen
sn: Jensen
uid: bjensen
streetAddress: 1 University Street
streetAddress;lang-en: 1 University Street
streetAddress;lang-fr: 1 rue University
preferredLanguage: fr
```

Users accessing this directory entry with an LDAP client with the preferred language set to English will see the address 1 University Street. Users accessing the directory with an LDAP client with the preferred language set to French will see the address 1 rue University.

Chapter

3

Extending the Directory Schema

Netscape Directory Server comes with a standard *schema* that includes hundreds of object classes and attributes. While the standard object classes and attributes should meet most of your requirements, you may need to extend your schema by creating new object classes and attributes.

This chapter describes how to extend your schema in the following sections:

Overview of Extending Schema

When you add new attributes to your schema, you must create a new object class to contain them. Although it may seem convenient to just add the attributes you need to an existing object class that contains most but not all of the attributes you require, doing so compromises the compatibility of your directory server with existing LDAP clients that rely on the standard LDAP schema and causes difficulties when upgrading your server. For the same reasons, you cannot delete standard schema elements.

For more information on object classes, attributes, and the directory schema as well as guidelines for extending your schema, refer to the *Netscape Directory Server Deployment Manual*. For information on standard attributes and object classes, see the *Netscape Directory Server Schema Reference Guide*.

To extend the directory schema you will:

1. Create new attributes. See “Creating Attributes” on page 66 for more information.
2. Create an object class to contain the new attributes and add the attributes to the object class. See “Creating Object Classes” on page 61 for more information.

Turning Schema Checking On and Off

When schema checking is on, the directory server ensures that the object classes and attributes you are using are defined in the directory schema, and that the attributes required for an object class are contained in the entry, and that only attributes allowed by the object class are contained in the entry.

Schema checking is turned on by default in the directory server, and Netscape recommends you run the directory server with schema checking turned on.

To turn schema checking on and off:

1. On the Directory Server Console, select the Configuration tab.
2. Select the root node in the navigation tree in the left pane and then select the Settings tab in the right pane.
3. To enable schema checking, select the “Enable Schema Checking” checkbox; clear it to turn off schema checking.
4. Click Save.

You can also turn schema checking on and off by using the Schema Checking parameter in `slapd.conf`. For information, see “Schema Checking” on page 465.

Managing Object Classes

You use the Directory Server Console to manage your schema's object classes. Through it, you can view all of your schema's object classes and create, edit, and delete your object class extensions to the schema. The following sections describe how to manage object classes:

For information on managing attributes, see "Managing Attributes" on page 64.

Viewing Object Classes

To view information about all object classes that currently exist in your directory schema:

1. On the Directory Server Console, select the Configuration tab.
2. Select the Database icon in the navigation tree in the left pane.
3. Select the Schema folder and then select the Object Classes tab in the right pane.
4. In the Object Classes list, select the object class you want to view.

   This tab displays information about the standard or user-defined object class you selected as defined in Table 3.1 on page 60.

Table 3.1 Object Classes tab reference

Table Header	Description
Parent	The parent identifies the object class from which this object class inherits its attributes and structure. For example, the parent object for the `inetOrgPerson` object class is the `organizationalPerson` object. That means that an entry with the object class `inetOrgPerson` must also include the object class `organizationalPerson`. Typically, if you want to add new attributes for use with user entries, the parent would be the `inetOrgPerson` object class. If you want to add new attributes for use with corporate entries, the parent is usually `organization` or `organizationalUnit`. If you want to add new attributes for use with group entries, the parent is usually `groupOfNames` or `groupOfUniqueNames`.
OID	The object identifier of the object class. An OID is a string, usually of dotted decimal numbers, that uniquely identifies an object, such as an object class or an attribute. If you do not specify an OID, the directory server automatically uses `<ObjectClass name>-oid`. For example, if you create the object class `division` without supplying an OID, the directory server automatically uses `division-oid` as the OID. For more information about OIDs, or to request a prefix for your enterprise, send mail to the IANA (Internet Assigned Number Authority) at iana@iana.org or visit the IANA website at: `http://www.iana.org/iana/`.
Object Classes	This list contains all of the standard and user-defined object classes in the directory server schema.
Required Attributes	Contains a list of attributes that must be present in entries using the object class.
Allowed Attributes	Contains a list of attributes that may be present in entries using the object class.

Creating Object Classes

You create an object class by giving it a unique name, selecting a parent object for the new object class, and adding required and optional attributes.

To create an object class:

1. On the Directory Server Console, select the Configuration tab.
2. Select the Database icon in the navigation tree in the left pane.
3. Select the Schema folder. and then select the Object Classes tab in the right pane.
4. Click Create.

   The Create Object Class dialog box appears.
5. Enter a unique name for the object class in the Name text box.
6. (Optional) Enter an object identifier for the new object class in the OID (Optional) text box.

   OIDs are described in Table 3.1 on page 60.
7. Select a parent object for the object class from the Parent pull-down menu.

   You can choose from any existing object class. See Table 3.1 on page 60 for more information on parent object classes.
8. To add an attribute that *must* be present in entries using the new object class: highlight the attribute in the Available Attributes list and then click the Add button to the left of the Required Attributes box. You can either use the standard attributes or create new ones. For information, see “Managing Attributes” on page 64.
9. To add an attribute that *may* be present in entries using the new object class: highlight the attribute in the Available Attributes list and then click the Add button to the left of the Allowed Attributes box.

10. To delete an attribute that you previously added, highlight the attribute in the Required Attributes list or the Allowed Attributes list and then click the corresponding Remove button.

11. Click OK when you have finished identifying the new object class and the required and allowed attributes.

Editing Object Classes

You can use the Server Console to edit object classes that you previously created. You cannot edit a standard object class.

To edit an object class:

1. On the Directory Server Console, select the Configuration tab.

2. Select the Database icon in the navigation tree in the left pane.

3. Select the Schema folder and then select the Object Classes tab in the right pane.

4. Select the object class you want to edit in the Object Classes list and click Edit.

   The Edit Object Class dialog box appears.

5. To change the name of the object class, enter the new name in the Name text box.

6. To change the object identifier for the object class, enter the new OID in the OID (Optional) text box.

   OIDs are described in Table 3.1 on page 60.

7. To change the parent object for the object class, select the new parent from the Parent pull-down menu.

8. To add an attribute that must be present in entries using the new object class, highlight the attribute in the Available Attributes list and then click the Add button to the left of the Required Attributes box. You can either use the standard attributes or create new ones. For information, see “Managing Attributes”.

9. To add an attribute that may be present in entries using the new object class, highlight the attribute in the Available Attributes list and then click the Add button to the left of the Allowed Attributes box.

10. To remove an attribute that you previously added, highlight the attribute in the Required Attributes list or the Allowed Attributes list and then click the corresponding Remove button.

11. Click OK when you are finished editing object classes.

Deleting Object Classes

You can delete only object classes that you have created. You cannot delete standard object classes.

To delete an object class:

1. On the Directory Server Console, select the Configuration tab.

2. Select the Database icon in the navigation tree in the left pane.

3. Select the Schema folder and then select the Object Classes tab in the right pane.

4. Select the object class you want to remove and click Delete.

5. If prompted, confirm the delete.

   The server immediately deletes the object class. There is no undo.

Managing Attributes

Through the Directory Server Console, you can view all attributes in your schema and you can create, edit, and delete your attribute extensions to the schema. The following sections describe how to manage attributes:

For information on managing object classes, see "Managing Object Classes" on page 59.

Viewing Attributes

To view information about all attributes that currently exist in your directory schema:

1. On the Directory Server Console, select the Configuration tab.
2. Select the Database icon in the navigation tree in the left pane.
3. Select the Schema folder and then select the Attributes tab in the right pane.

   This tab contains information about all the standard (read-only) and user-defined attributes in the schema as defined in Table 3.2 on page 65.

Table 3.2 Attributes tab reference

Table Header	Description
Name	The name of the attribute.
OID	The object identifier of the attribute. An OID is a string, usually of dotted decimal numbers, that uniquely identifies an object, such as an object class or an attribute. If you do not specify an OID, the directory server automatically uses `<attribute name>-oid`. For example, if you create the attribute `birthdate` without supplying an OID, the directory server automatically uses `birthdate-oid` as the OID. For more information about OIDs, or to request a prefix for your enterprise, send mail to the IANA (Internet Assigned Number Authority) at iana@iana.org or visit the IANA website at: `http://www.iana.org/iana/`.
Syntax	The attribute syntax: • Case Ignore String—Indicates that values for this attribute are not case sensitive. • Case Exact String—Indicates that values for this attribute are case sensitive. • Distinguished Name—Indicates that values for this attribute are DNs. • Binary—Indicates that values for this attribute are binary. • Telephone—Indicates that values for this attribute are in telephone format. • Integer—Indicates that valid values for this attribute are numbers. • Operational—Operational attributes are not returned as a result of an ldapsearch operation unless they are explicitly specified in the search. Generally, operational attributes are reserved for use by the directory server.
Multi	If the attribute is multivalued, an X appears in this column, otherwise, the server leaves this field blank. The directory server allows more than one instance of a multi-valued attribute per entry.

Creating Attributes

You use the Directory Server Console to create new attributes. Whenever you want to add new attributes to your schema, you must create a new object class to contain them. See “Creating Object Classes” on page 61 for more information.

To create a new attribute:

1. On the Directory Server Console, select the Configuration tab.
2. Select the Database icon in the navigation tree in the left pane.
3. Select the Schema folder and then select the Attributes tab in the right pane.
4. Click Create. The Create Attribute dialog box appears.
5. Enter a unique name for the attribute in the Attribute Name text box.
6. (Optional) Enter an object identifier for the attribute in the Attribute OID (Optional) text box.

   OIDs are described in Table 3.2 on page 65.
7. Select a syntax that describes the data to be held by the attribute from the Syntax pull-down menu.

   Available syntaxes are described in Table 3.2 on page 65.
8. If you want the attribute to be multi-valued, select the Multi-Valued checkbox. The Directory Server allows more than one instance of a multivalued attribute per entry.
9. Click OK.

Editing Attributes

You can edit only attributes you have created. You cannot edit standard attributes.

To edit an attribute:

1. On the Directory Server Console, select the Configuration tab.
2. Select the Database icon in the navigation tree in the left pane.
3. Select the Schema folder and then select the Attributes tab in the right pane.
4. Select the attribute you want to edit in the User Defined Attributes table and click Edit.

   The Edit Attribute dialog box appears.
5. To change the attribute's name, enter a new one in the Attribute Name text box.
6. To change the attribute's object identifier, enter a new one in the Attribute OID (Optional) text box.

   OIDs are described in Table 3.2 on page 65.
7. To change the syntax that describes the data to be held by the attribute, choose a new one from the Syntax pull-down menu.
8. Available syntaxes are described in Table 3.2 on page 65.
9. To make the attribute multivalued, select the Multi-Valued checkbox. The Directory Server allows more than one instance of a multivalued attribute per entry.
10. When you have finished editing the attribute, click OK.

Deleting Attributes

You can delete only attributes that you have created. You cannot delete standard attributes.

To delete an attribute:

1. On the Directory Server Console, select the Configuration tab.
2. Select the Database icon in the navigation tree in the left pane.
3. Select the Schema folder.

   The schema configuration tabs appear in the right pane.
4. Select the Attributes tab.
5. In the User Defined Attributes table, select the attribute and click Delete.
6. If prompted, confirm the delete.

   The server immediately deletes the attribute. There is no undo.

Chapter

4

Managing Directory Server Databases

The directory managed by your directory server is contained in a database. This chapter describes the procedures you use to manage your database in the following sections:

Managing Databases Using LDIF

You can use the LDAP Data Interchange Format (LDIF) to import and export database entries into and out of the directory server. You can also back up your databases by exporting to an LDIF file. These topics are described in:

For a description of LDIF, see Chapter 2, "LDAP Data Interchange Format."

Exporting Databases to LDIF

Exporting your database to LDIF can be useful if you want to back up your database, copy your database to another directory server, export your database to another application, or add an index setting for a currently existing attribute. You can export your database to LDIF either by using the Directory Server Console, or by using the `slapd` (Windows NT) or `ns-slapd` (Unix) command-line utility. This section describes both these methods in the following sections:

Exporting to LDIF Using the Server Console

To export your database to LDIF from the Directory Server Console:

1. On the Directory Server Console select the Configuration tab.
2. From the Console menu select Export.

   The Export Database dialog box displays.

3. If you are running the Directory Server Console on the server's host machine, skip to Step 4. Otherwise, if you want to export to a file on the local machine, select "Local machine". To export to a file on the server's host, select "Server machine."

4. Enter the full path and filename you want the server to use to store the LDIF file in the text box provided.

   Netscape recommends that you use the `<NSHOME>/slapd-<serverID>/ldif` directory on the server's host machine to store LDIF files.

   - If you chose "Local machine" in Step 3, enter the path relative to the machine where you are running the Directory Server Console. Otherwise, enter the path relative to the server's host machine.

   - If you chose to export a file to the local machine, or if you are running the Directory Server Console on the server's host machine, you can also click Browse to select the path and filename.

5. If you want to export the whole directory, select the "Entire database" radio button. If you want to export only a single suffix or a particular subtree, select the Subtree radio button and then enter the suffix or subtree you want to export in the Subtree text box. You can also click Browse to select a suffix or subtree.

6. Click OK.

Exporting to LDIF From the Command Line

You can export your database to LDIF using the `slapd` (Windows NT) or `ns-slapd` (Unix) command-line utility with the `db2ldif` keyword. For information on where you can find the command line utilities, see "Finding the Command-Line Utilities" on page 33.

Use one of the following syntaxes to export your database to LDIF from the command-line. Parameters in brackets [ ] are optional.

On Windows NT:

```
slapd db2ldif -f <slapd.conf> -a <output_file>
[-d <debug_level> -n -r -s <include_suffix>
-x <exclude_suffix> -u -g <generation_type>]
```

On Unix:

```
ns-slapd db2ldif -f <slapd.conf> -a <output_file>
[-d <debug_level> -n -r -s <include_suffix>
-x <exclude_suffix> -u -g <generation_type>]
```

where <*slapd.conf*> is the location of your configuration file. The `slapd.conf` file is under <*NSHOME*>`/slapd-`<*serverID*>`/config`. Enter the full path to the `slapd.conf` file you want to use.

ns-slapd and slapd Parameters for Exporting Databases

-a. Defines the output file in which the server saves the exported LDIF. This file is stored by default in the directory where the command-line utility resides.

-d. (Optional) Specifies the debug level to use during the `db2ldif` runtime. Debug levels are defined in “Log Level” on page 444.

-f. Specifies the `slapd.conf` configuration file to use for the conversion process. Use the full path to the `slapd.conf` file with this argument. For information on where to find directory server configuration files, see “Directory Server Configuration Files” on page 36.

-g. (Optional) By default, the server uses random number generation to assign an arbitrary (based on time) ID that is unique to each entry. You can override this behavior by specifying `-g` followed by a generation type. The server supports the following two generation types:

- none—The server will not generate unique IDs for entries.
- deterministic—The server considers the ID for entries to be named-based and will be the same if the same name is used and different otherwise. You would use deterministic generation based on the entry's DN.

Use `-g` if you want import the same LDIF file into two different directory servers and you want the contents of both directories to have the same set of unique IDs.

-n. (Optional) Specifies that entry IDs are not to be included in the LDIF output. The entry IDs are necessary only if the `db2ldif` output is to be used as input to `db2index`.

-r. (Optional) If you intend to import the LDIF file into a consumer server, you must specify this argument. `-r` causes the server to include the `copiedFrom` attribute and its contents in the LDIF output. The replication process requires this information. If you use `-r`, you also need to specify the suffix you want exported using the `-s` option. You must shut down the server before exporting using this option.

-s. (Optional) Specifies the suffix or suffixes to include in the export. You may use multiple `-s` arguments. This parameter is optional. If you do not specify `-s` or `-x`, the server exports all suffixes within the database. If you use both `-x` and `-s` arguments with the same suffix, the `-x` operation takes precedence. Exclusion always takes precedence over inclusion. If you exclude one or more suffixes from the exported LDIF file, and you intend to import the LDIF file into your configuration directory, do not exclude `o=NetscapeRoot`.

-u. (Optional) By default, the server includes the uniqueID in the exported LDIF file. Specify this parameter if you do not want the uniqueID included in the LDIF output. If you intend to use the exported LDIF to initialize a 4.x consumer server, you need to specify this parameter.

-x. (Optional) Specifies a suffix or suffixes to exclude in the export. You may use multiple `-x` arguments. This parameter is optional. If you do not specify `-s` or `-x`, the server exports all suffixes within the database. If you use both `-x` and `-s` arguments with the same suffix, the `-x` operation takes precedence. Exclusion always takes precedence over inclusion. If you exclude one or more suffixes from the exported LDIF file, and you intend to import the LDIF file into your configuration directory, do not exclude `o=NetscapeRoot`.

Database to LDIF Examples

Windows NT:

```
slapd db2ldif
-f c:\Netscape\Server4\slapd-dirserver\config\slapd.conf
-a output.ldif -s "o=airius.com"
```

Unix:

```
ns-slapd db2ldif
-f /usr/Netscape/Server4/slapd-dirserver/config/slapd.con
f -a output.ldif -s o=airius.com
```

Importing Databases From LDIF

You can import LDIF files into your database from the Directory Server Console or by using the `slapd` (Windows NT) or `ns-slapd` (Unix) command-line utility. This section describes both these methods.

When you import an LDIF file using `ldif2db` or by specifying the "Overwrite Entire Database" option (with "Preserve Configuration") through the Server Console, a snapshot of `o=NetscapeRoot` is saved to a file and stored in `<NSHOME>/slapd-<serverID>/confbak`. If you need to, you can restore the configuration information in the directory by importing the most recent file in this directory. The files are named according to the date on which the import took place as follows:

```
YYYY_MM_DD_HHMMSS.ldif
```

For example, if the file was saved at 42 seconds past 10 PM on July 30, 1998, it would be named:

```
1998_07_30_224244.ldif
```

In most cases, you will want to add the configuration information to the existing data instead of overwriting your existing data. To do this from the Directory Server Console, when you import the LDIF file, clear the "Overwrite Entire Database" option on the Import dialog box. You can also use `ldapmodify` as follows:

```
ldapmodify -D "<BindDN>" -w <BindDN password> -c -a
-f <NSHOME>/slapd-<serverID>/confbak/filename.ldif
```

For example,

```
ldapmodify -D "cn=directory manager"
-w mypassword -c -a -f
/Netscape/Server4/slapd-mydirserver/confbak/1998_07_30_22
4244.ldif
```

For better performance, you should use the server console to import an LDIF file only if the LDIF file contains a relatively small number of entries (less than 10,000), or if you are importing and overwriting the existing database. Otherwise, you should use the command line. This section describes:

- "Importing LDIF From the Server Console" on page 75
- "Importing LDIF From the Command Line" on page 77

Importing LDIF From the Server Console

You can use the Directory Server Console to import the LDIF file into a directory server database using the Import command. For best performance, you should use the server console to import an LDIF file only if the LDIF file contains a relatively small number of directory entries (less than 10,000), or if you intend to overwrite the existing database. Otherwise, you should use the command-line. For more information, see "Importing LDIF From the Command Line" on page 77.

Note You cannot import an LDIF file that contains a root entry unless you bind to the directory as the Root DN (Directory Manager). This is because, access to the root entry, for example `o=airius.com`, is denied to everyone except the Directory Manager (Root DN).

To import LDIF using the Directory Server Console:

1. On the Directory Server Console select the Configuration tab.
2. From the Console menu, select Import. This displays the Import Database dialog box.
3. If you are running the Directory Server Console on the server's host machine, skip to Step 4. Otherwise, if you want to import a file from the local machine, select "From local machine". If you want to import a file from the server's host, select "From server machine."
4. Enter the full path to the LDIF file in the field provided.

   If you chose to import a file from the local machine, or if you are running the Directory Server Console on the server's host machine, you can also click Browse to select the file you want to import.

5. Select the import method you want the server to use. There are two options:

   Overwrite Entire Database. You can only choose this option if you chose to import a file on the server's host (in Step 3) or if you are running the Directory Server Console on the server's host machine. When you import using this option, the server deletes the entire contents of the database and imports the LDIF file. If you do not want the server to overwrite the `o=NetscapeRoot` suffix, select the Preserve Configuration checkbox.

   If the directory server is running, you are prompted to shut it down.The server must be shut down before you can import using this option.

   WARNING! The Netscape Administration Server uses the `o=NetscapeRoot` suffix to store information about installed Netscape Servers. Deleting this suffix could force you to reinstall all of your Netscape 4.x servers, including the directory server. Netscape strongly recommends you choose to preserve this configuration unless directed otherwise by Netscape Technical Support or other procedures outlined in the directory server documentation.

   Append Data to Database. When you import using this option, the server does not delete the contents of the directory before adding the entries from the LDIF file. You should only use this option if you are importing an LDIF file with a relatively small number of entries (less than 10,000). The server must be running to use this option.

   You cannot import an LDIF file that creates a root entry (such as `o=airius.com`) using the "Append Data to Database" option unless you bind to the directory as the root DN, for example, `cn=Directory Manager`. Instead, you must use the "Overwrite Entire Database" option.

   The optional settings you can specify include:

   - **Add Only.** LDIF file may contain modify and delete instructions in addition to the default add instructions. If you want the server to ignore operations other than add, select the "Add only" checkbox.

   - **Continue on Error.** If you want the server to continue with the import even if errors occur, select the "Continue on error checkbox". You might want to use this option if you are importing an LDIF file that contains

some entries that are already in the database in addition to new ones. The server notes existing entries in the rejects file (but otherwise ignores them) while adding all new entries.

- **Read / Values From Files.** If you want the server to interpret values that begin with a forward slash "/" or a drive letter "C:\" as file names, select the Read / values from files checkbox. If you select this option, the contents of these files, rather than the file names, will be stored in the directory.

- **File for Rejects.** The server keeps a record of all entries that it cannot import. This might happen, for example, if an entry already exists in the database or if there is no parent object for the entry you are trying to add. If you leave this field blank, the server will not record rejects. By default, the server stores the rejects file in the same directory where the LDIF file you are importing is stored. If you want, you can specify a full path where you want the server to store the file.

6. Click OK.

The server performs the import and also creates indexes. (For more information on indexes and index creation, refer to Chapter 7, "Managing Indexes.")

Importing LDIF From the Command Line

You can create a new `ldbm` database file from an LDIF file using the `slapd` (Windows NT) or `ns-slapd` (Unix) command-line utility. These utilities create the database in the location specified in the "directory" parameter in your `slapd.ldbm.conf` file and create the index files that are specified in the `slapd.ldbm.conf` "Attribute to be Indexed" parameter.

For more information, see the "Database" parameter and the "Attribute to be Indexed" parameter in Chapter 17, "Configuration Parameters."

For information on where you can find the command-line utilities in your directory server installation, see "Finding the Command-Line Utilities" on page 33.

To import LDIF from the command line:

1. From the command line, change to <*NSHOME*>/slapd-<*serverID*>/db. Where <*NSHOME*> is the directory where you installed the directory server and <*serverID*> is the name of your directory server. The directory server database must be stored in this directory.

2. Make a backup of all the files in the db directory. Although you can delete the files, you may want to move them to a backup location instead, because deleting these files deletes your directory database.

3. Change to <*NSHOME*>/bin/slapd/server.

4. Run the slapd (Windows NT) or ns-slapd (Unix) command-line utility as follows. Parameters in brackets [ ] are optional.

   Windows NT:

   ```
   slapd ldif2db -f <slapd.conf> -C -i <ldif_file>
   [-d <debug_level> -n <backend_number> -O
   -s <include_suffix> -x <exclude_suffix>]
   ```

   Unix:

   ```
   ns-slapd ldif2db -f <slapd.conf> -C -i <ldif_file>
   [-d <debug_level> -n <backend_number> -O
   -s <include_suffix> -x <exclude_suffix>]
   ```

   where <*ldif_file*> is the name of the file containing the LDIF to be imported and <*slapd.conf*> is the location of your configuration file. You can find a demo LDIF file under <*NSHOME*>/slapd-<*serverID*>/ldif. The slapd.conf file is under <*NSHOME*>/slapd-<*serverID*>/config. Enter the full path to the slapd.conf file you want to use.

slapd Parameters Used for LDIF Imports

The following ldif2db parameters are used to complete an LDIF file import:

-C. Required. Used internally by the Directory Server.

-d. Optional. Specifies the debug level to use during runtime. Debug levels are defined in “Log Level” on page 444.

-f. Specifies the `slapd.conf` file to use for the import process. This parameter is required. For information on where to find directory server configuration files, see “Directory Server Configuration Files” on page 36.

-i. Specifies the LDIF file to be imported. This parameter is required. You can use multiple `-i` arguments to import more than one LDIF file at a time. When you import multiple files, the server imports the LDIF files in the order in which you specify them from the command line.

-n. Optional. Specifies the database in your `slapd.conf` file for which the conversion is performed. If this parameter is not specified, then the server uses the first database defined in the `slapd.conf` file.

-O. Optional. When you use this argument, no attribute indexes are created for the imported database. If you specify this option and you want to restore the indexes later, you will need to recreate the indexes by hand. See Chapter 7, “Managing Indexes,” for more information.

-s. Optional. Specifies the suffix or suffixes within the LDIF file you want to import. If you exclude one or more suffixes from the LDIF file (using `-x`), and you are importing this LDIF file into the configuration directory, use `-s` to ensure that `o=NetscapeRoot` is included in the import. You can use multiple `-s` arguments. If you use both `-x` and `-s` with the same suffix, `-x` takes precedence. Exclusion always takes precedence over inclusion. If you do not specify `-x` or `-s`, then all available suffixes will be imported from the LDIF file.

-x. Optional. Allows you to specify suffixes within the LDIF file to exclude during the import. You can use multiple `-x` arguments. This option lets you selectively import portions of the LDIF file. If you use both `-x` and `-s` with the same suffix, `-x` takes precedence. Exclusion always takes precedence over inclusion. If you do not specify `-x` or `-s`, then all available suffixes will be imported from the LDIF file.

Warning! If you are importing the LDIF file into your configuration directory, do not exclude the suffix `o=NetscapeRoot`. The Netscape Administration Server uses this suffix to store information about installed Netscape Servers. Deleting this suffix could force you to reinstall all of your Netscape 4.x servers, including the directory server.

LDIF to Database Examples

Windows NT:

```
slapd ldif2db -f c:\Netscape\Server4\slapd-dirserver\config\slapd.conf
-i c:\Netscape\Server4\slapd-dirserver\ldif\demo.ldif -i
c:\Netscape\Server4\slapd-dirserver\ldif\demo2.ldif -s "o=Airius.com"
-x "o=NetscapeRoot"
```

Unix:

```
ns-slapd ldif2db -f
/usr/Netscape/Server4/slapd-dirserver/config/slapd.conf
-i /usr/Netscape/Server4/slapd-dirserver/ldif/demo.ldif
-i /usr/Netscape/Server4/slapd-dirserver/ldif/demo2.ldif
-s "o=Airius.com" -x o=NetscapeRoot
```

Deleting LDIF Files

If you want, you can delete LDIF files you have created. The Directory Server Console does not provide functionality to do this. Instead, you need to delete the files from the command line or through your operating system's utilities.

Backing Up and Restoring Your Database

You can back up and restore your database from the Directory Server Console. To back up your database, you can perform an online backup using the Directory Server Console or the `db2bak` command-line script. You can also manually copy your database directly to a backup directory.

When restoring your database, you must shut down your server. However, you can back up your database while the server is running.

The following sections describe the options available to you:

- "Deleting Database Backups" on page 84
- "Restoring Databases That Include Replicated Entries" on page 84

Exporting your database to and importing from LDIF is described in "Exporting Databases to LDIF" on page 70 and "Importing Databases From LDIF" on page 74.

Backing Up Your Database From the Server Console

When you back up your database from the Directory Server Console, the server copies the entire database and associated index files to a backup location.

To perform an online backup of your database from the server console:

1. On the Directory Server Console select the Tasks tab.
2. Click "Back Up the Directory Server". The Backup directory dialog box appears.
3. Choose a directory name where you want the backup stored in one of two ways: type in the name of the directory in which you want the backup placed in the Directory text box, or click "Use default" and the server provides a name for the backup directory.

   If you choose to use the default, the backup files will be placed in the following location:

   <*NSHOME*>/slapd-<*serverID*>/bak/<*backup_directory*>

   where <*backup_directory*> is a directory given the name of the backup. By default, the backup directory name identifies the time and date when the backup was created in the format `YYYY_MM_DD_HHMMSS`.
4. Click OK.

Backing Up Your Database From the Command Line

You can back up your database from the command line by using the `db2bak` command-line script. This script assumes you are using the `slapd.conf` file located in `<NSHOME>/slapd-<ServerID>/config`. Where *<NSHOME>* is the directory where you installed the directory server and *<serverID>* is the name of your directory server.

To perform an online backup of your directory from the command line:

1. At the command prompt, change to `<NSHOME>/slapd-<serverID>`.
2. Run the `db2bak` command-line script as follows:

   `db2bak [backup_directory]`

   You can choose to specify a full path or just a directory where you want the server to store the backup. If you only specify a directory, the server creates the directory under `<NSHOME>/slapd-<ServerID>/`.

   If you do not specify either a full path or a single directory, the script makes a copy of your database and stores it in `<NSHOME>/slapd-<ServerID>/bak/<backup_directory>`. Where *<backup_directory>* is a directory given the name of the backup. By default, the backup directory name identifies the time and date when the backup was created in the format `YYYY_MM_DD_HHMMSS`.

Restoring Your Database From the Server Console

If your database becomes corrupted, you can restore from a previously generated backup using the Directory Server Console. This process consists of copying the database and associated index files from the backup location to the database directory. See “Backing Up Your Database From the Server Console” on page 81 for more information.

WARNING! Restoring your database overwrites your existing database files, if any.

To restore your database from a previously created backup:

1. On the Directory Server Console select the Tasks tab.
2. Click “Restore Directory Server”. The Restore Directory dialog box displays.
3. The Console lists all backups in the default directory (<*NSHOME*>/slapd-<*serverID*>/bak/<*backup_name*>) in the Available Backups list box.

   You can either select the backup from this list or enter the full pathname to a location containing a valid backup in the Directory text box.
4. Click OK.

   If the server is running, you are prompted to shut it down. The restore cannot continue while the server is running.

Restoring Your Database From the Command Line

You can restore your database from the command line by using the `bak2db` command-line script. This script assumes you are using the `slapd.conf` file located in <*NSHOME*>/slapd-<*ServerID*>/config. Where <*NSHOME*> is the directory where you installed the directory server and <*serverID*> is the name of your directory server.

To restore your directory from the command line:

1. At the command prompt, change to <*NSHOME*>/slapd-<*serverID*>.
2. If the server is running, type `stop-slapd` to shut it down.
3. Run the `bak2db` command-line script as follows:

   `bak2db [`*`backup_directory`*`]`

Deleting Database Backups

By default, the server console places backup files that it creates in a directory under <*NSHOME*>/slapd-<*serverID*>/bak. If you want to remove old backups, you need to delete the files from this directory using the command line or through your operating system's utilities.

Restoring Databases That Include Replicated Entries

If you are restoring a database that is supplying entries to other servers, then you must reinitialize all of your consumer servers. A message will be logged to the consumer servers' log files indicating that reinitialization is required. If you want reinitialization to occur automatically, you can modify the ORCauto parameter. See Chapter 17, "Configuration Parameters," for information.

If you are restoring a database containing data received from a supplier server, then one of two situations can occur:

- Change log entries have not yet expired on the supplier server. If change log entries have not expired on the supplier server since the local database backup was taken, then you can simply restore the local consumer and continue with normal operations. This situation is likely to occur only if the backup was taken within a period of time that is shorter than the value you have set for the Max Changelog Age parameter in slapd.conf.
- Change log entries have expired on the supplier server since the time of the local backup. In this case, the consumer server will automatically be reinitialized.

For information on managing replication, see Chapter 13, "Managing Replication." For information on initializing consumers, see "Initializing Consumers" on page 344.

Placing a Database in Read-Only Mode

You must put a database in read-only mode if you are manually initializing a consumer. For information on manually initializing a consumer, see “Initializing Consumers” on page 344.

When a database is in read-only mode you cannot create, modify, or delete any entries.

If your directory server manages multiple databases, you can place all of them into read-only mode at the same time by placing your entire server in read-only mode. For instructions on how to do this, see “Placing the Entire Directory Server in Read-only Mode” on page 294.

If you want to place a database into read-only mode from the command line, set the `slapd.conf` Read-only parameter to `on`. You must shut the server down before you edit the configuration files.

To place a database into read-only mode from the server console:

1. On the Directory Server Console select the Configuration tab.
2. Select the Database icon in the navigation tree in the left pane.
3. Select the Settings tab in the right pane.
4. Select the “Make Database Read-Only” checkbox.
5. Click Save.

Setting Suffixes for Your Database

Your directory server can simultaneously manage many different directory trees. Each directory tree is represented by a suffix, and each suffix corresponds to the root (or topmost) entry in the directory. When the directory server receives a request, the server checks its list of suffixes against the request to see if the server is managing the directory tree the client wants to access. If the directory request does not match the directory trees the server is managing, then the server sends the client a referral, if one has been configured. If a referral has not been configured, the server returns an error.

Your Netscape Directory Server always uses multiple suffixes. However, most of these are used only for internal purposes, and only one corresponds to the primary directory that you are using the server to manage (such as `o=Airius.com`). Other suffixes correspond to directory trees used internally by the server (such as `o=NetscapeRoot`, or the change log).

You can add, modify, and delete suffixes for your database. If you delete an existing suffix, then LDAP clients will not be able to access the entries represented by that suffix.

For information on the Suffix parameter, see “Suffix” on page 484.

To manage suffixes for your server:

1. On the Directory Server Console select the Configuration tab.
2. Select the Database icon in the navigation tree. This displays the database settings in the right pane.
3. Select the Settings tab. This tab contains a list of all the current suffixes in your directory.
4. To add a new suffix, click Add and enter the new suffix in the field that appears.

   If the suffix value contains a comma, you must precede the comma with a backslash (\). For example, to add Airius Bolivia, S.A. as a suffix, you would enter `Airius Bolivia\, S.A.` in the Suffix field.
5. To delete a suffix, select it in the list and click Delete.

   WARNING! Do not delete the suffix `o=NetscapeRoot`. The Netscape Administration Server uses this suffix to store information about installed Netscape Servers. Deleting this suffix could force you to reinstall all of your Netscape 4.x servers, including the directory server.
6. To modify an existing suffix, double-click the suffix in the list and make your changes.
7. Click Save.

Enabling and Disabling Plug-Ins From the Server Console

You can enable and disable plug-ins over LDAP using the directory server console. To do this:

1. On the Directory Server Console, select the Configuration tab.
2. Double-click the Plugins folder in the navigation tree.
3. Select the plug-in in the plug-ins list.
4. To disable the plug-in, clear the “Enabled” checkbox. To enable the plug-in, select this checkbox.
5. Click Save.
6. Restart the directory server.

Managing the Referential Integrity Plug-in

Referential integrity is a database mechanism that ensures that relationships between related entries are maintained. In the Directory Server, referential integrity can be used to ensure that a directory update to one entry is correctly reflected in any other entries that may refer to the updated entry.

For example, if a user’s entry is removed from the directory and referential integrity is enabled, the server also removes the user from any groups of which the user is a member. If referential integrity is not enabled, the user remains a member of the group until manually removed by the administrator. This is an important feature if you are integrating the directory server with other Netscape products that rely on the directory for user and group management.

Whenever you delete or rename a user or group entry in the directory, the operation is logged to the referential integrity log file (`<NSHOME>/slapd-<serverID>/logs/referint`). After a specified time, known as the *update interval*, the server searches the directory for all attributes that have been set for integrity updates that have a DN equal to the value of the

deleted or modified entries. If the log file shows that the entry was deleted, the corresponding attribute is deleted. If the log file shows that the entry was changed, the corresponding attribute value is modified accordingly.

To maintain referential integrity in a replicated environment, you should configure the plug-in on the supplier to record any changes made (due to integrity updates) in the change log. You should also disable the referential integrity plugins on all consumer servers. The supplier server sends any changes made by the referential integrity plug-in to consumer servers. It is therefore unnecessary to run the referential integrity plug-in on consumer servers, unless your consumer servers master data locally and you want to maintain referential integrity within that locally-mastered data.

By default, the referential integrity plug-in is enabled and set to perform integrity updates on the `member`, `uniquemember`, `owner`, and `seeAlso` attributes immediately after a delete or rename operation. You can; however, disable the plug-in if you do not need this feature, configure the plug-in to record changes in the change log, change the update interval, and choose the attributes you want the plug-in to update. The rest of this section explains how in the following sections:

- "Managing Referential Integrity From the Server Console" on page 88
- "Managing Referential Integrity From the Command Line" on page 89
- "Configuring Referential Integrity for Replicated Environments" on page 90
- "Changing the Integrity Update Interval" on page 91
- "Modifying Which Attributes to Update" on page 92

Managing Referential Integrity From the Server Console

You can enable or disable the Referential Integrity Postoperation plug-in from the Directory Server Console as described in "Enabling and Disabling Plug-Ins From the Server Console" on page 87.

Managing Referential Integrity From the Command Line

You can enable or disable the referential integrity plug-in from the command line by editing the `plugin postoperation` parameter in the `slapd.ldbm.conf` file (for information on the location of the configuration files, see “Directory Server Configuration Files” on page 36).

To enable or disable the plug-in:

1. Stop the server. See “Starting and Stopping the Directory Server” on page 29 for information.

2. Open the `slapd.ldbm.conf` file and locate the line that begins:

   On Windows NT:

   ```
   plugin postoperation on “referential integrity
   postoperation” “<NSHOME>/lib/referint-plugin.dll”
   referint_postop_init
   ```

   On Unix:

   ```
   plugin postoperation on “referential integrity
   postoperation” <NSHOME>/lib/referint-plugin.so
   referint_postop_init
   ```

3. To disable the plug-in, change the integer value immediately after `referint_postop_init` to `-1`. For example:

   ```
   plugin postoperation
   “<NSHOME>/lib/referint-plugin.dll”
   referint_postop_init “-1”
   “<NSHOME>/slapd-<serverID>/logs/referint” “0”
   member uniquemember owner seeAlso
   ```

   To enable the plug-in, change the `-1` back to zero (0).

4. Save the file.

5. Start the server.

   See “Starting and Stopping the Directory Server” on page 29 for information.

Configuring Referential Integrity for Replicated Environments

From the command-line, you can configure the plug-in on a supplier server to record any changes it makes in the change log. The supplier server will then send any changes made by the referential integrity plug-in to consumer servers. If you configure the plug-in on the supplier server to maintain referential integrity for replication, you do not need to enable referential integrity on its consumers.

To configure the plug-in to record changes in the change log:

1. Stop the server. See "Starting and Stopping the Directory Server" on page 29 for information.

2. Open the `slapd.ldbm.conf` file and locate the line that begins:

   On Windows NT:

   ```
   plugin postoperation on "referential integrity
   postoperation" "<NSHOME>/lib/referint-plugin.dll"
   referint_postop_init
   ```

   On Unix:

   ```
   plugin postoperation on "referential integrity
   postoperation" <NSHOME>/lib/referint-plugin.so
   referint_postop_init
   ```

3. Change the integer value immediately after `"<NSHOME>/slapd-<serverID>/logs/referint"` to `1`. For example:

   ```
   plugin postoperation
   "<NSHOME>/lib/referint-plugin.dll"
   referint_postop_init "0"
   "<NSHOME>/slapd-<serverID>/logs/referint" "1"
   member uniquemember owner seeAlso
   ```

   To disable this feature, change the `1` back to zero (0).

4. Save the file.

5. Start the server.

   See “Starting and Stopping the Directory Server” on page 29 for information.

Changing the Integrity Update Interval

By default, the referential integrity plug-in searches the database and updates related entries immediately after a delete or rename operation. If you want to reduce the impact this operation has on your system, you may want to increase the amount of time between updates. Although there is no maximum update interval, the following intervals are commonly used:

- Update immediately
- 90 seconds (updates occur every 90 seconds)
- 3600 seconds (updates occur every hour)
- 10,800 seconds (updates occur every 3 hours)
- 28,800 seconds (updates occur every 8 hours)
- 86,400 seconds (updates occur once a day)
- 604,800 seconds (updates occur once a week)

To modify the update interval:

1. Stop the server.
2. Open the `slapd.conf` file and locate the line that begins

   ```
   plugin postoperation <NSHOME>/lib/referint-plugin.dll
   referint_postop_init
   ```

3. Change the integer value that immediately follows `referint_postop_init` to the number of seconds between updates.

For example, if you want to change the update interval so that updates occur once a day (every 86,400 seconds), you edit the line as follows:

```
plugin postoperation
"<NSHOME>/lib/referint-plugin.dll"
referint_postop_init 86400
"<NSHOME>/slapd-<serverID>/logs/referint" "0"
member uniquemember owner seeAlso
```

4. Save the file.

5. Start the server. See "Starting and Stopping the Directory Server" on page 29 for information.

Modifying Which Attributes to Update

By default, the referential integrity is set up to update the `member`, `uniquemember`, `owner`, and `seeAlso` attributes. You can either add or delete attributes to be updated by editing the `slapd.ldbm.conf` file (for information on the location of configuration files, see "Directory Server Configuration Files" on page 36). For best performance, the attributes set for updating should also be indexed. For information on indexing, see Chapter 7, "Managing Indexes."

To modify which attributes should be updated:

1. Open the `slapd.ldbm.conf` file and locate the line that begins

   ```
   plugin postoperation
   "<NSHOME>/lib/referint-plugin.dll"
   referint_postop_init
   ```

2. Add or delete attribute names from the end of the line that begins

   ```
   plugin postoperation
   "<NSHOME>/lib/referint-plugin.dll"
   referint_postop_init
   ```

For example, to modify the plug-in to perform integrity updates on the `manager` attribute, add `manager` to the end of the plug-in postoperation line as follows:

```
plugin postoperation
“<NSHOME>/lib/referint-plugin.dll”
referint_postop_init 0 “<NSHOME>/slapd-<serverID>
/logs/referint” “0” member uniquemember
owner seeAlso manager
```

3. Save the file.

4. Restart the server.

Managing Database Transaction Logging

Whenever a directory database operation such as a write is performed, the server logs the operation by default to the transaction log. For best performance, the operation itself may not be performed immediately. Instead, it is stored in a temporary memory cache on the directory server until the operation is completed. If the server experiences a failure, such as a power outage, and shuts down abnormally, the information about recent directory changes that were stored in the cache are lost. However, when the directory server restarts, it automatically detects the error condition and uses the database transaction log file to recover the database.

Although database transaction logging and database recovery are automatic processes that require no intervention, you may want to tune some of the database transaction logging parameters for best performance.

The following sections describe the parameters:

- “Changing the Location of the Database Transaction Log” on page 94
- “Changing the Database Checkpoint Interval” on page 94
- “Disabling Durable Transactions” on page 95

Changing the Location of the Database Transaction Log

By default, the database transaction log file is stored in the `<NSHOME>/slapd-<serverID>/db` directory along with the directory files themselves. Because the purpose of the transaction log is to aid in the recovery of a directory database that was shut down abnormally, it is a good idea to store the database transaction log on a different disk from the one containing the directory database. Storing the database transaction log on a separate physical disk may also improve directory server performance.

You can move the location of the database transaction log file by adding the `db_logdirectory` parameter to the end of the `slapd.ldbm.conf` file.

For information on the location of the configuration files, see “Directory Server Configuration Files” on page 36. For information on the `db_logdirectory` parameter syntax, see “Database Transaction Log Directory” on page 479.

Changing the Database Checkpoint Interval

Whenever a directory database operation such as a write or modify is performed, the operation is logged to the directory server database transaction log. For best performance, the results of the operation itself may not be written to disk immediately. Instead they are stored in a temporary memory cache on the directory server. At specific intervals, the directory server writes the previously cached data out to the disk and logs a checkpoint entry in the database transaction log. By indicating which changes have already been written to the directory, checkpoint entries tell the directory server where in the database transaction log to begin recovery, thus speeding up the recovery process.

By default, the directory server is set up to send a checkpoint entry to the database transaction log every 60 seconds. Increasing the checkpoint interval may increase the performance of directory server write operations. Increasing the checkpoint interval may also significantly increase the amount of time required to recover the directory database after a disorderly shutdown and may

waste disk space due to overly large database transaction log files. Therefore, you should only modify this parameter if you are familiar with database optimization and can fully assess the impact of the change.

To modify the checkpoint interval, you must add the `db_checkpoint_interval` parameter to the end of the `slapd.ldbm.conf` file.

For information on the location of the configuration files, see “Directory Server Configuration Files” on page 36. For information on the `db_checkpoint_interval` parameter syntax, see “Database Checkpoint Interval” on page 476.

Disabling Durable Transactions

By default, durable database transaction logging is enabled. This means that every time a write is performed on the directory, a corresponding entry is physically written to the database transaction log disk. To improve performance, you can disable durable transaction logging. When you do so, every directory database operation is logically written to the database transaction log file, but it may not be physically written to disk immediately. That means that if a directory change was written to the logical database transaction log file but not physically written to disk at the time of a system crash, you cannot recover the change. When durable transactions are disabled, the recovered database is consistent, but does not reflect the results of any LDAP write operations that completed just before the system crash.

You can disable durable transactions by adding the `db_durable_transactions` parameter to the end of the `slapd.ldbm.conf` file and set its value to `off`.

For information on the location of the configuration files, see “Directory Server Configuration Files” on page 36. For information on the `db_durable_transactions` parameter syntax, see “Database Durable Transactions” on page 478.

Chapter

5

Managing Access Control

Netscape Directory Server provides you with the ability to control access to your directory. This chapter describes the directory server access control mechanism, a feature that is both powerful and flexible, you will want to spend some time planning your security policy before you start setting permissions for your directory.

This section includes the following topics:

Refer to the *Netscape Directory Server Deployment Manual* for tips on planning your access control strategy.

Understanding Access Control

The mechanism by which you define access is called *access control*. When the server evaluates an incoming request, it determines access based on the access control instructions (ACIs) you define. The collection of ACIs within a single suffix is called an access control list (ACL). The server uses the information in the ACIs to allow or deny permissions such as read, write, search and compare.

Using access control, you can set permissions for the entire directory, for a subset of the directory, for specific entries in the directory, for a specific set of entry attributes, or for configuration tasks for any 4.x Netscape Server. In addition, you can set permissions for a specific user, all users belonging to a specific group, or all users of the directory. Finally, you can define access for a specific location such as an IP address or a DNS name.

Each entry in the directory can contain one or more ACI attributes, which holds the access control information for the entry. The access control instruction (ACI) is composed of three parts:

Targets. The target specifies what object, object attributes, or group of objects and attributes you are controlling access to.

Permissions. The permission specifically outlines what rights you are either allowing or denying.

Bind Rules. The bind rules specify the circumstances under which access is to be granted or denied. Bind rules indicate who can access the directory, when the directory can be accessed, and the physical network locations that the directory can be accessed from.

The permission and bind rule portions of the ACI are set as a pair, also called an Access Control Rule (ACR), and you can have multiple permission-bind rule pairs for each target. This allows you to efficiently set multiple access controls for a given target. For example:

```
<target>(<permission><bind rule>)(<permission><bind rule>)...
```

The following sections describe targets, permissions, and bind rules in more detail.

Targets

The target identifies what directory entry the ACI applies to. You can target a directory entry (usually a branch of your directory tree), a directory entry and one or more entry attributes, or a group of entries and/or attributes that are the result of a single LDAP filter. Each method of targeting is detailed in the following sections:

- "Targeting a Directory Entry" on page 99
- "Targeting Attributes" on page 100
- "Targeting Using LDAP Filters" on page 100

Targeting a Directory Entry

When you target a directory entry, the ACIs you set apply to the target and all of its children. For example, if you target the entry `ou=accounting, o=airius.com`, the ACI will apply to all entries in the accounting branch of the Airius tree. Most often, you will want to place your ACIs on branch points in the directory rather than on individual leaf objects.

You can also use wildcards when targeting a directory entry. For example, targeting `uid=c*a, o=airius.com` would target all directory entries with user ID attributes that start with `c` and end with `a`. You cannot use wildcards in the suffix portion of the distinguished name. For example, targeting `uid=bjensen, o=*.com` is invalid.

If you are setting access control using the Directory Server Console, you target an entry by selecting it on the Directory tab. For more information, see "Setting Access Control Using the Server Console" on page 110. If you are setting access control using LDIF, "Setting Targets Using LDIF" on page 137.

Keep in mind that the entry you target must be at the same level or a child of the entry containing the ACI. For example, if you are modifying an ACI attribute that resides on the `ou=accounting, o=airius.com` entry, you cannot

target the `uid=sarette, ou=people, o=airius.com` entry because it is not a child of the accounting tree; rather it is within a separate branch of the Airius tree.

Targeting Attributes

In addition to targeting directory entries, you can also target one or more attributes included in the targeted entry. This is useful when you want to deny or allow access to partial information about an entry. For example, you could allow access to only the common name, surname, and telephone number attributes of a given entry. Or you could deny access to sensitive information such as passwords or salary information. If you do not specify any attributes, then you are setting access control on the entry itself, not on particular attributes contained within the entry. For example, you might use this to provide add and delete access rights to an entire entry, instead of access rights to particular attributes contained within the entry.

You can specify that the target attribute either is (=) or is not (!=) equal to a specific attribute. The attributes you supply should be recognized members of your schema, although they do not have to be allowed by the object class of the targeted entry. For a listing of the attributes in Netscape's standard schema, see the *Netscape Directory Server Schema Reference Guide*.

If you are setting access control using the server console, you specify one or more attributes to deny or allow access to using the Set Access Permissions dialog box. For more information, see "Setting Access Control Using the Server Console" on page 110. If you are setting access control using LDIF, see "Using the targetattr Keyword" on page 139.

Targeting Using LDAP Filters

In addition to explicitly targeting directory entries and attributes, you can also use LDAP filters to target a group of entries and/or attributes that match a certain criteria. When searching the directory, LDAP filters select the entries to be returned as a result of the search operation. The same is true when you use LDAP filters in ACIs, except that the entries and attributes returned are then targeted by the ACI.

For example, if your target is `o=airius.com`, you could set the filter to target the `ou=people` and `ou=groups` trees. Without the filter, the ACI would apply to all entries below `o=airius.com`. However, with the filter, the ACI will

only apply to the `ou=people, o=airius.com` and `ou=groups, o=airius.com` entries and their child entries. For more information on using LDAP search filters, see Chapter 8, “Finding Directory Entries.”

If you are setting access controls using the server console, you can specify a target filter in the Target Filter area of the Select Attributes dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

If you are setting access control using LDIF files, see “Using the targetfilter Keyword” on page 140.

Although targeting using LDAP filters can be useful when you are targeting entries and attributes that are spread across the directory, the results are sometimes unpredictable. Because search filters do not directly name the object that you are managing access for, it is easy to unintentionally allow or deny access to the wrong objects, especially as your directory becomes more complex. Additionally, filters can make it difficult for you to troubleshoot ACI problems within your directory. If you do use filtering, do so sparingly. Be sure to test your directory access thoroughly each time you change a filter so that you are certain what the results of the change mean for access to your directory.

Permissions

Permissions specify the type of access you are allowing or denying. You can either allow or deny an entity from performing specific operations to the directory. The various operations that can be assigned are known as rights.

If you selected Typical Install during installation, then by default, all members of the Administrators group and the user defined in the `Root DN` parameter have unlimited access to the directory. However, if no `aci` attributes exist within the directory then only the user defined in the `Root DN` parameter has unlimited access. By default all other users are denied access rights of any kind.

The user defined in the `Root DN` parameter is known as the root or unrestricted user. The unrestricted user has full access to your directory regardless of the permissions set for the directory. For this reason, you must set some permissions for your directory if you want any normal users to be able to access your directory. There are two parts to setting permissions: allowing or denying access, and assigning rights. These two parts are described next.

Allowing or Denying Access

You can either explicitly allow or deny access to your directory tree. When an LDAP client attempts to perform any kind of access to a directory entry, the directory server looks for access control information from the entry being accessed back to the top, or root, of the directory tree. When deciding whether to allow or deny access, you should keep the following precedence rule in mind.

Precedence Rule. If two permissions exist and are in conflict, the permission that denies access always takes precedence over the permission that grants access.

For example, if you deny write permission at the directory's root level, and you make that permission applicable to everyone accessing the directory, then no user can write to the directory regardless of the specific permissions you grant that user. To allow a specific user write permissions to the directory, you have to restrict the scope of the original denial for write permission so that it does not include the user. Then you have to create an additional allow for write permission for the user in question.

Because of this, and because by default users are denied access anyway, you should use deny permissions sparingly in order to avoid confusion.

For more guidelines on when to deny and when to allow access, refer to the *Netscape Directory Server Deployment Manual.*

When setting access control using the server console, you specify whether to allow or deny access using the Set Access Permissions dialog box. For more information, see "Setting Access Control Using the Server Console" on page 110.

When setting access control using LDIF, you must use the allow or deny keywords in the permission portion of the ACI statement to explicitly allow or deny access. For more information, see "Setting Permissions Using LDIF" on page 140.

Assigning Rights

Rights detail the specific operations a user can perform on directory data. You can allow or deny all rights, or you can assign one or more of the following rights:

Read. Indicates whether directory data may be read.

Write. Indicates whether attributes may be added, modified, or deleted.

Add. Indicates whether the user or application can create entries.

Delete. Indicates whether entries can be deleted.

Search. Indicates whether the directory data can be searched for. Users must have Search and Read rights in order to view the data returned as part of a search operation.

Compare. Indicates whether the data may be used in comparison operations. With compare rights, the directory returns a yes or no in response to an inquiry, but the user cannot see the value of the entry or attribute.

Selfwrite. Indicates whether people can add or delete themselves from a group. This right is only used for group management.

Proxy. Indicates whether the specified entry can access the target with the rights of another entry. See “Overview of Proxied Authorization” on page 158 for more information.

All. Indicates that the specified entry has all rights (read, write, search, delete, compare, and selfwrite) to the targeted entry excluding proxy.

When setting access control using the server console, you specify which rights to allow or deny using the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

When setting access control using LDIF, you use the rights keywords within the permission portion of the ACI statement to explicitly allow or deny access. For more information, see “Setting Permissions Using LDIF” on page 140.

Bind Rules

Binding refers to logging in or authenticating to the directory. The circumstances under which binding occurs determine whether access to the directory is allowed or denied. Every permission set in an ACI has a corresponding bind rule that details the specific circumstance under which binding must occur for the ACI to be applied.

Bind rules can be simple. For example, a bind rule can simply state that the person accessing the directory must belong to a specific group. Bind rules can also be more complex. For example, a bind rule can state that a person must belong to a specific group and must log in from a machine with a specific IP address, between 8 AM and 5 PM.

Whether access is allowed or denied depends on whether an ACI's bind rule is evaluated to be true. Bind rules use one of the two following patterns:

```
<keyword> = "<expression>";
```

```
<keyword> != "<expression>";
```

where equal (=) indicates that `<keyword>` and `<expression>` must match in order for the bind rule to be true, and not equal (!=) indicates that `<keyword>` and `<expression>` must not match in order for the bind rule to be true.

Bind rules define who can access the directory, when, and from where. More specifically, bind rules specify

- users and groups that can access the directory
- location from which an entity must bind
- time or day on which binding must occur
- type of authentication that must be in use during binding

Additionally, bind rules can be complex constructions that combine bind methods using Boolean operators. See "Boolean Bind Rules" for more information.

Each bind method is detailed in the following sections.

User and Group Access

Most commonly, bind rules state who can access the directory. A bind rule may state any of the following:

- Anyone can access the targeted resource. This is known as *Anonymous Access*.
- All authenticated users can access the targeted resource. This is known as *General Access.*
- A specific user can access the targeted resource. This is known as *User Access*.
- Members of a specific group can access the targeted resource. This is known as *Group Access*.
- A user or group specified in an attribute of another user entry can access the targeted resource. This is known as *Access Based on Attribute Value*.

These five types of user and group access are described in the following sections.

Anonymous Access

Anonymous access can be configured for the directory such that anyone can access it. In this situation, users do not need to provide a bind DN or password to gain access. You can limit anonymous access to specific types of access (for example, access for read or access for search) or to specific subtrees or individual entries within the directory.

If you are setting access control using the server console, you define anonymous access through the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

For information on setting anonymous access using LDIF, see “Using the userdn Keyword” on page 143.

General Access

You can use bind rules to indicate that the permission applies to anyone who has successfully bound to the directory, that is, all authenticated users. This allows general access while preventing anonymous access.

If you are setting access control using the server console, you define general access on the Set Access Permissions dialog box. For more information, see "Setting Access Control Using the Server Console" on page 110.

For information on setting up general access using LDIF, see "Using the userdn Keyword" on page 143.

User Access

You can use bind rules to specify that access to the targeted resource will be granted or denied only if the user binds using a specific DN. You can also specify groups of users by using the wildcard character (*). For example, specifying a user DN of `uid=u*, o=airius.com` indicates that only users with a bind DN beginning with the letter `u` will be allowed or denied access based on the permissions you set.

If you are setting access control using the server console, you set user access from the Set Access Permissions dialog box. For more information, see "Setting Access Control Using the Server Console" on page 110.

For information on setting user access using LDIF, see "Using the userdn Keyword" on page 143.

Using LDAP URLs. You can dynamically target users in ACIs using a URL with a filter as follows:

```
"ldap:///<suffix>??sub?(filter)"
```

For example, all users in the accounting and engineering branches of the Airius tree would be granted or denied access to the targeted resource dynamically based on the following URL:

```
"ldap:///o=airius.com??sub?(ou=engineering)(ou=accounting)"
```

Note Do not specify a hostname or port number within the LDAP URL or the server will skip the URL.

For more information about LDAP URLs, see Appendix A, "LDAP URLs."

Parent Access. Another special user access bind rule is the case in which a user is granted or denied access to the entry only if the bind DN is the parent of the targeted entry.

If you are setting access control using the server console, you set up parent access on the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

For information on setting parent access using LDIF, see “Using the userdn Keyword” on page 143.

Self Access. Another special user access bind rule is the case in which you want to grant or deny users access to their own entries. In this case, access would be granted or denied if the bind DN matches the DN of the targeted entry.

If you are setting access control using the server console, you set up self access on the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

For information on setting self access using LDIF, see “Using the userdn Keyword” on page 143.

Group Access

You can use bind rules to specify that access to a targeted entry will be granted or denied only if the user binds using a DN that belongs to a specific group.

If you are setting access control using the server console, you define specific groups on the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

If you are setting access control using LDIF, see “Using the groupdn Keyword” on page 144 for information on setting group access.

Access Based on Attribute Value

You can set a bind rule such that the ACI applies only if the bind DN matches a DN in a specific attribute of the targeted entry. The named attribute must be one that is expected to contain a full DN. For example, you can specify that the bind DN must match the DN in the manager attribute of a user entry in order for the ACI to apply. In this situation, only the user's manager would have access to the entry.

If you are setting access control using the server console, you use the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

If you are setting access control using LDIF, see “Using the userdnattr and groupdnattr Keywords” on page 144 for information on setting user and group DN attribute access.

Access From a Specific Machine or Domain

Using bind rules, you can indicate that the ACI is applicable only if the bind operation is arriving from a specific IP address or fully qualified domain name or hostname. This is often used to force all directory updates to occur from a given machine or network domain.

You can also use the wildcard character (*) to denote multiple machines. For example, you could use a wildcard IP address such as 12.3.45.* to specify a specific subnetwork or 123.45.6.*+255.255.255.115 to specify a subnetwork mask. Or you could use a wildcard domain name such as `*.airius.com` to specify a specific DNS domain.

If you are setting access control using the server console, you can define specific machines to which the ACI applies through the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

If you are setting access control using LDIF, see “Using the ip Keyword” on page 147 and “Using the dns Keyword” on page 148 for information on defining machine-based access.

Access at a Specific Time of Day or Day of Week

You can also use bind rules to specify that the ACI will only apply at a certain time of day or on a certain day of the week. For example, you can set a rule that will allow access only if it is between the hours of 8 AM and 5 PM Monday through Friday. The time used is the time on the local host.

If you are setting access control using the server console, you can define specific access times through the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

If you are setting access control using LDIF, see “Using the timeofday Keyword” on page 148, and “Using the dayofweek Keyword” on page 149.

Access Based on Authentication Method

You can also set bind rules that state that the ACI will apply only if a client binds to the directory using a specific authentication method. Here are the authentication methods you specify:

None. Authentication is not required. This is the default.

Simple. The ACI will apply only if the client accesses the directory using a user name and password.

SSL. The ACI will apply only if the client accesses the directory over a Secure Sockets Layer (SSL) connection (over LDAPS). For more information on setting up SSL, see Chapter 11, “Managing SSL.”

SASL. The ACI will apply only if the client accesses the directory over a Simple Authentication and Security Layer (SASL) connection. For information on setting up SASL, see the *Netscape Directory Server Programmer’s Manual*.

If you are using the server console to set access control, you can set up authentication-based bind rules through the Set Access Permissions dialog box. For more information, see “Setting Access Control Using the Server Console” on page 110.

If you are using LDIF to set up access control, see “Using the authmethod Keyword” on page 149 for information on setting authentication-based bind rules.

Boolean Bind Rules

Bind rules can be complex expressions that use the Boolean expressions `AND`, `OR`, and `NOT` to set very precise access rules. If you are using the server console to set access control, you will need to use the Extra area (or Customized Expressions) to enter the LDIF commands if you want to use boolean bind rules. For more information, see “Using Boolean Expressions in LDIF Bind Rules” on page 150.

Boolean expressions are evaluated in the following order:

- innermost to outermost parenthetical expressions first
- all expressions from left to right

Also, the Boolean `OR` and Boolean `AND` operators have no order of precedence, and they are evaluated from left to right. However, the boolean `NOT` operator is evaluated before the Boolean `OR` and Boolean `AND`. Thus, in the following example the `NOT` operator is evaluated before the `AND` operator:

```
<bind rule 1> and not <bind rule 2>
```

Setting Access Control Using the Server Console

You can use the Directory Server Console to create, edit, and delete access control for your directory. You may set up multiple ACIs for a given entry in the directory. Although you can combine access control rules (ACRs) in an access control instruction (ACI) for a particular entry, Netscape recommends that you create a separate ACI for each ACR.

Although the procedures you perform will depend on the specific access control required for your organization based on the security policy you have developed, this section provides general instructions for setting access control in the following sections:

- “Creating a New ACI” on page 111
- “Editing an Existing ACI” on page 117
- “Deleting an Existing ACI or ACR” on page 117

See “Access Control Usage Examples” on page 118 for a collection of some access control rules commonly used in directory server security policies, along with step-by-step instructions for using the Directory Server Console to create these standard ACIs.

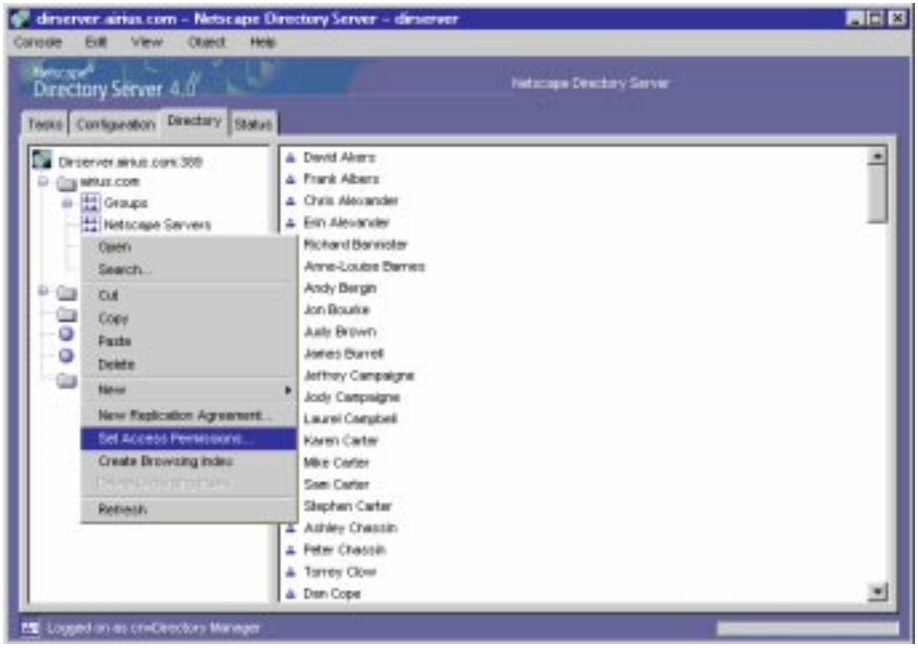
Creating a New ACI

To create a new ACI:

1. Make sure the directory server is running.
2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to ACIs that have been set for the directory. See “Binding to the Directory From Netscape Console” on page 27 for specific instructions.
3. On the Directory Server Console, select the Directory tab.
4. Right-click the entry in the navigation tree for which you want to set access control, and select Set Access Permissions from the pop-up menu (Figure 5.1). The Multi-value ACI Selector dialog box appears.

Figure 5.1 Selecting an object in the navigation tree to set access control



5. Click New.

   The Set Access Permissions dialog box appears.

   The table lists the access control rules (ACRs) defined for this ACI. By default, the first ACR in the table denies access to everyone with the exception of the root DN (Directory Manager).

6. Click ACI Attributes.

   The Select Attributes dialog appears.

7. (Optional) If you want to change the name of the ACI, type the new name in the ACI Name text box.

   The name can be any string you want to use to uniquely identify the ACI. If you do not enter a name, the server uses “unknown”.

8. If you want to change the target, you may do so now. The Target text box contains the DN of the entry you selected in the Directory navigation tree.

   The target must be either the DN of the currently selected entry, or a direct or indirect child of the currently selected entry. Remember, the ACI you define applies to the target entry and all subentries in the directory tree. If you use the suffix, for example, `o=airius.com` as the target, the ACI applies to the entire directory tree.

   The target must be a valid DN. You can use the default “is” to set a target that is equal to the DN you enter or select “is not” to set a target that is not equal to the DN you enter.

   If the DN you target contains a comma as part of its value, you must precede the comma with a single backslash (\) escape character.

9. You may enter a search filter in the Target Filter text box.

   You can use the default “is” to set a target filter that is equal to the value you enter or select “is not” to set a target filter that is not equal to the value you enter.

10. You may enter an attribute to target in the Target Attribute text box.

By default, all attributes (*) are targeted. You can use the default “is” to set a target attribute that is equal to the value you enter or select “is not” to set a target attribute that is not equal to the value you enter. If you want to enter more than one attribute, separate the attributes with a double-pipe “||”. Click OK to return to the Set Access Permissions dialog box.

11. To check or modify the LDIF syntax of the ACI, click View/Edit Syntax on the Set Access Permissions dialog box.

The Edit ACI Syntax dialog box displays.

For more information about modifying ACIs in LDIF, see “Setting Access Control Using LDIF Files” on page 135. When you are finished, click OK to return to the Set Access Permissions dialog box.

12. To edit an ACR in the table, double-click the cell to display a dialog box for entering additional information. Cells and related options are summarized in Table 5.1. For more specific information about access control parameters, refer to the online help.

Table 5.1 The ACI Editor settings and options

Setting	What it does	Options
Add Rule	Adds an ACR to the ACI.	
Delete Rule	Deletes the currently selected ACR from the ACI.	
Show Inherited Rules	Displays rules inherited from branchpoints in the directory tree superior to the current entry.	On/Off.
Rule number	Describes the order in which the ACRs were created.	You cannot modify the rule number. This number has no effect on ACR precedence. Refer to “Allowing or Denying Access” for information about ACR precedence.

Table 5.1 The ACI Editor settings and options (Continued)

Setting	What it does	Options
Allow/Deny	Specifies whether to grant or restrict access to the resources named in this rule.	Double-click the cell in the Set Access Permissions dialog box and choose Allow or Deny from the drop-down list.
Host	Designates host computers affected by this rule.	**DNS Hostname.** Enter a host name or names, use commas to separate multiple entries, and then click Add to add them to the list. **IP Address.** Enter the hosts IP address or use the wildcard * to define multiple hosts. You can only use the wildcard * at the end of an IP address. The * must replace an entire byte in the address. For example, 198.95.251.* is acceptable; 198.95.251.3* is unacceptable. Click Add to add them to the list. **All hosts except those on this list.** To exclude listed hosts from the rule, select this option.
Time	Specifies an interval when the rule will be in effect.	Define the start and end time in 24 hour format (HHMM). For example, `1400` signifies 2PM.

Table 5.1 The ACI Editor settings and options (Continued)

Setting	What it does	Options
User/Group	Designates users or groups affected by this rule.	In the Select Users & Groups dialog box: **Anyone.** Default. Allows unlimited anonymous access. **Users or groups.** To add a user or group to the list, select either "Add user to list" or "Add group to list" from the drop down, enter user or group names in the text box, and click Add. You can use wildcard patterns to add multiple users or groups at one time. **All users/groups except those specified in the list.** To exclude listed users and groups from the rule, select this option. **User DN Attribute.** Requires an attribute that is expected to contain a full distinguished name. For example, `manager`. See "Using the userdn Keyword" for information. If you want to use groupdnattr, you must define the ACI using LDIF. See "Setting Access Control Using LDIF Files" for information. **Authentication Method.** The options include: • None—Select None to indicate that no authentication is required for the ACR to apply. Use this in conjunction with Anyone (see above). This is the default. • Simple—Select "Simple" to indicate that the ACR will only apply if the client binds using simple (user name and password) authentication. • SSL—Select SSL to indicate that the ACI will only apply if the client binds using an SSL connection. • SASL External—Select this option if you have written a Directory Server plug-in for use with SASL authentication.

Table 5.1 The ACI Editor settings and options (Continued)

Setting	What it does	Options
Rights	Specifies rights allowed or denied by this rule.	**Read.** User can view an entry. **Write.** User can change or delete an entry. **Search.** Indicates whether data can be searched for. Users must have Search and Read rights in order to view the data returned as part of a search operation. **Compare.** Indicates whether data may be used in comparison operations. With compare rights, the directory returns a yes or no in response to an inquiry, but the user cannot see the value of the entry or attribute. **Selfwrite.** Indicates whether people can add or delete themselves from a group. This right is only used for group management. **Delete.** User can delete entries. **Add.** User can add entries. **Proxy.** User can authenticate with the rights of another entry in order to modify the target entry. See “Overview of Proxied Authorization” on page 158 for more information. **All.** User has all access rights to the entry except proxy.
View/Edit Syntax	Lets you view or edit the LDIF ACI syntax.	For more information on editing ACIs using LDIF, see “Setting Access Control Using LDIF Files”.

13. If you want, you can add more than one ACR to the ACI. To do so, click New Rule and edit as necessary.

    Netscape recommends that you create a new ACI for each ACR in your directory. Doing so increases performance and manageability.

14. When you are finished editing the ACI, click OK. The server creates the new ACI.

Editing an Existing ACI

To edit an existing ACI, do the following:

1. Make sure the directory server is running.
2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See “Binding to the Directory From Netscape Console” on page 27 for specific instructions.
3. On the Directory Server Console, select the Directory tab.
4. Right-click the entry in the navigation tree for which you want to edit access control, and select Set Access Permissions from the pop-up menu.

   A dialog appears prompting you to select the ACI you want to edit. Select the ACI and click OK. The Set Access Permissions dialog box appears.

   The Set Access Permissions dialog box contains the ACRs and other information about the ACI. For details on the information you can edit using this dialog box, see Table 5.1.
5. Make the desired changes to the various areas of the Set Access Permissions dialog box.
6. Click OK when you have finished editing the ACI.

Deleting an Existing ACI or ACR

To delete an ACI or ACR, do the following:

1. Make sure the directory server is running.
2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See “Binding to the Directory From Netscape Console” on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the entry in the navigation tree from which you want to remove the ACI, and select Set Access Permissions from the pop-up menu.

   A dialog appears prompting you to select an ACI.

   If you want to delete an entire ACI, select the ACI you want to delete and click Delete. You are done.

   If you only want to remove an ACR from the ACI:

   - select the ACI in the list and click OK. The Set Access Permissions dialog box appears.
   - Select the ACR in the table and click Delete Rule and then click OK. The ACR is deleted immediately. There is no undo.

Access Control Usage Examples

The following examples describe how to set some of the more common directory permissions using the Directory Server Console:

- Setting Anonymous Access for Read, Search, and Compare
- Allowing Users to Modify Their Own Directory Entries
- Allowing Users to Change Some of Their Own Attributes
- Granting a Group Full Access to a Suffix
- Granting a Group Rights to Add and Delete Entries
- Allowing Full Access to a Specific Branch Point
- Allowing Access at a Specific Time of Day or Day of Week
- Allowing Updates Only From a Specific Location
- Allowing Access to a Suffix Over SSL Only
- Setting a Target Using Filtering
- Allowing Users to Add or Remove Themselves From a Group

Setting Anonymous Access for Read, Search, and Compare

Most directories are run such that you can anonymously access at least one suffix for read, search, or compare. For example, you might want to set these permissions if you are running a corporate personnel directory that you want employees to be able to search, such as a phonebook. You can set this permission by doing the following:

1. Make sure the directory server is running.
2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.
3. On the Directory Server Console, select the Directory tab.
4. Right-click the suffix entry in the navigation tree, for example, `o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.
5. Click New.

   The Set Access Permissions dialog box appears.
6. Click ACI Attributes.

   The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as "`anonymous access`". In the future, this will allow you to see at a glance which of your access controls allows anonymous read and search. Click OK to return to the Set Access Permissions dialog box.
7. Click the cell under Allow/Deny in the table and select Allow from the drop down menu.

8. Double-click the cell under Rights in the table.

   The Select Rights dialog box appears. Select the checkboxes next to Read, Search, and Compare, and deselect all the other checkboxes on the dialog box. Click OK when you are finished to return to the Set Access Permissions dialog box.

9. Click OK.

   The server saves the ACI.

Allowing Users to Modify Their Own Directory Entries

Many directories allow users to change attribute values in their own entries. The following procedure shows you how to set this permission.

Note By setting this permission, you are also granting users the right to delete attribute values.

1. Make sure the directory server is running.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See “Binding to the Directory From Netscape Console” on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the suffix entry in the navigation tree, for example, `o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes.

   The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `allow-self-write`. In the future, this will allow you to see at a glance which of your access controls allows users to edit their own entries. Click OK to return to the Set Access Permissions dialog box.

7. Double-click the cell under User/Group in the table.

   The Select Users and Groups dialog box appears. Select Add User to List from the drop-down menu, type `self` in the text box provided, and then click Add. Click OK to return to the Set Access Permissions dialog box.

8. Double-click the cell under Rights in the table. Select the Write checkbox and click OK to return to the Set Access Permissions dialog box.

9. Click OK.

   The server saves the new ACI.

Allowing Users to Change Some of Their Own Attributes

Many directory administrators want to allow users to change some but not all of the attributes in their own entry. For example, you may want to allow a user to change his or her own password or telephone number, but nothing else. The following procedure shows you how to set this permission.

Note By setting this permission, you are also granting users the right to delete attribute values.

1. Make sure the directory server is running.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See “Binding to the Directory From Netscape Console” on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the suffix entry in the navigation tree, for example, `o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes.

   The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `write pw and tel`. In the future, this will allow you to see at a glance which of your access controls allows users to edit their own passwords and telephone numbers.

7. In the Target Attribute(s) field, enter `userpassword || telephonenumber`.

8. Click OK to return to the Set Access Permissions dialog box.

9. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

10. Double-click the cell beneath User/Group in the table.

    The Select Users and Groups dialog box displays. Select Add User to List from the drop-down menu, type `self` in the text box provided, and then click Add. Click OK to return to the Set Access Permissions dialog box.

11. Double-click the cell under Rights in the table. The Select Rights dialog box displays. Select the Write checkbox, and then click OK to return to the Set Access Permissions dialog box.

    This allows users to only write their own attributes. If you want the users to be able to read, search, or compare these attribute values, then make sure you set a separate permission to allow this. If you have set up anonymous search in your directory as described in “Setting Anonymous Access for Read, Search, and Compare” on page 119, then you do not have to create any further permissions.

12. Click OK.

    The server saves the new ACI.

Granting a Group Full Access to a Suffix

Most directories have a group that is used to identify directory administrators. This group of users is usually given full access to all or part of the directory. By applying the access rights to the group, you can avoid setting the access rights for each directory administrator individually. Instead, you grant users these access rights simply by adding them to the group.

The following procedure shows you how to grant the members of an administrators group full access to the user directory under the suffix `o=airius.com`.

You can create groups and add or delete members from the group using the directory tab on the Directory Server Console. See Chapter 9, "Managing Directory Entries," for information. In addition, you can also add entries using the directory server gateway, or the Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* and the online help available through the gateway for information.

1. Make sure the directory server is running and that you have created a group whose members include the directory administrators.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the suffix entry in the navigation tree, for example, `o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes. The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `directory admin all`.

   In the future, this will allow you to see at a glance which of your access controls allows the directory administrators group full access to the directory. Click OK to return to the Set Access Permissions dialog box.

7. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

8. Double-click the cell under User/Group in the table.

   The Select Users and Groups dialog box appears. Select Add Group to List from the drop-down menu, type the DN for the directory administrators group in the text box provided. For example, `cn=administrators, o=airius.com`, and then click Add. Click OK to return to the Set Access Permissions dialog box.

9. Click OK.

   The server saves the new ACI.

Granting a Group Rights to Add and Delete Entries

Many directories have a group that is used to identify individuals who add and delete entries from the directory regularly, such as a Human Resources group that is responsible for adding and deleting employees from the directory. By applying access rights to the group, you can avoid setting the access for each group member individually. Instead, you grant users these access rights simply by adding them to the group.

The following procedure shows you how to grant the members of the HR administrators group add and delete access to the user directory under the `o=airius.com` suffix.

You can create groups and add or delete members from the group using the directory tab on the Directory Server Console. See Chapter 9, “Managing Directory Entries,” for information. In addition, you can also add entries using

the directory server gateway, or the Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* and the online help available through the gateway for information.

1. Make sure the directory server is running and that you have created a group whose members include the HR managers.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the suffix entry in the navigation tree, for example, `o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes. The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `HR managers all`.

   In the future, this will allow you to see at a glance which of your access controls allows the HR managers group full access to the user directory. Click OK to return to the Set Access Permissions dialog box.

7. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

8. Double-click the cell under User/Group in the table. The Select Users and Groups dialog box appears. Select Add Group to List from the drop-down menu, type the DN for the HR managers group in the text box provided. For example, `cn=HR managers, o=airius.com`., and then click Add. Click OK to return to the Set Access Permissions dialog box.

9. Double-click the cell under Rights in the table. The Select Rights dialog box displays. Select the Add and Delete checkboxes, and then click OK to return to the Set Access Permissions dialog box.

10. Click OK.

    The server saves the new ACI.

Allowing Full Access to a Specific Branch Point

One type of access control that is commonly used is to set up specific administrators of individual subdirectories. This allows you, for example, to have a group of people that are responsible for administering an Accounting subtree and another group of people that are responsible for administering a Marketing subtree.

The following procedure shows you how to allow a group called accounting administrators to have full directory access to the accounting subtree.

Before you can set these permissions, you must create the accounting branch point (`ou=accounting, o=airius.com`). You can create organizational unit branch points in your directory using the directory tab on the Directory Server Console. See Chapter 9, "Managing Directory Entries," for information. In addition, you can also add entries using the directory server gateway, or the Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* and the online help available through the gateway for information.

1. Make sure the directory server is running and that you have created a group whose members include the accounting managers. For example, `cn=accounting managers, ou=accounting, o=airius.com`.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the accounting entry in the navigation tree, for example, `ou=accounting, o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes. The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `Accounting managers all`. In the future, this will allow you to see at a glance which of your access controls allows the accounting managers group full access to the accounting branch of the directory. Click OK to return to the Set Access Permissions dialog box.

7. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

8. Double-click the cell under User/Group in the table. The Select Users and Groups dialog box appears. Select Add Group to List from the drop-down menu, type the DN for the accounting managers group in the text box provided. For example, `cn=accounting managers, ou=accounting, o=airius.com`, and then click Add. Click OK to return to the Set Access Permissions dialog box.

9. Click OK.

   The server saves the new ACI.

Allowing Access at a Specific Time of Day or Day of Week

Many directories have “blackout” periods where users cannot write to the directory. When setting up time-based access restrictions, it may be easier to manage an ACI that explicitly restricts time-based access rather than to search though the directory for all ACIs that allow “write” and restricting their scopes to exclude access during the designated blackout period. The following

example shows you how to set up access control so that all write access to the directory (under a single suffix, `o=airius.com`) is denied between the hours of 11 am and 1 pm Sunday.

1. Make sure the directory server is running.
2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.
3. On the Directory Server Console, select the Directory tab.
4. Right-click the suffix entry in the navigation tree, for example, `o=airius.com`, and select Set Access Permissions from the pop-up menu. The Multi-value ACI Selector dialog box appears.
5. Click New.

   The Set Access Permissions dialog box appears.
6. Click ACI Attributes. The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `time/date`. In the future, this will allow you to see at a glance which of your access controls restricts access to the directory based on the time of day and day of week. Click OK to return to the Set Access Permissions dialog box.
7. Click the cell beneath Allow/Deny in the table and select Deny from the pull-down menu.
8. Double-click the cell beneath Time in the table. The Select Times dialog box appears. In the Beginning at text box, type the beginning time in 24 hour format without the colon ":", for example, 1100. In the Ending at text box, enter the end time (1300). Type in the day(s) you want the ACR to be in effect. Enter the first three characters of the day you want to enter, for example, sun, mon, tue, wed, etc. Separate multiple days with commas (,). Click OK to return to the Set Access Permissions dialog box.
9. Click OK.

   The server saves the new ACI.

Allowing Updates Only From a Specific Location

Many directories restrict directory updates to clients running from a specific DNS hostname or IP address. This ensures that the directory cannot be updated unless the person has access to a specific machine. This type of update restriction is most common when you are populating your directory using some HR or accounting package, or by using some kind of LDAP gateway.

The following procedure shows you how to allow write access to the directory server running on `accounting.airius.com` only to clients running on the machine named `abacus.airius.com`.

Before you can set these permissions, you must create the accounting branch point (`ou=accounting, o=airius.com`). You can create organizational unit branch points using the directory tab on the Directory Server Console. See Chapter 9, “Managing Directory Entries,” for information. In addition, you can also add entries using the directory server gateway, or the Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* and the online help available through the gateway for information.

1. Make sure the directory server is running and that you have created a group whose members include the accounting managers. For example, `cn=accounting managers, ou=accounting, o=airius.com`.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See “Binding to the Directory From Netscape Console” on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the accounting entry in the navigation tree, for example, `ou=accounting, o=airius.com`, and select Set Access Permissions from the pop-up menu.The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes. The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `Updates from abacus`. In the future, this will allow you to see at a glance which of your access controls allows updates from the machine names abacus. Click OK to return to the Set Access Permissions dialog box.

7. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

8. Double-click the cell under Host in the table. The Select Hosts and IP Addresses dialog box appears. Select "DNS Hostname" from the drop-down menu, type the fully-qualified domain name of the host in the text box provided, for example, `abacus.airius.com`, and then click Add. Click OK to return to the Set Access Permissions dialog box.

9. Click OK.

   The server saves the new ACI.

Allowing Access to a Suffix Over SSL Only

You might want the directory data being read or updated to be encrypted so that it cannot be captured or tampered with. This will often be the case if you are allowing updates to your directory from outside your firewall. For example, the directory administrator may want to be able to update or query the user directory from home.

The following procedure shows you how to allow access from a machine with the IP address `123.45.6.78` outside the `airius.com` firewall only if the user binds to the directory using a Secure Sockets Layer (SSL) connection.

1. Make sure the directory server is running.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the suffix entry in the navigation tree, for example, `o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes. The Select Attributes dialog displays. In the ACI Name text box, enter a unique value such as `SSL`. In the future, this will allow you to see at a glance which of your access controls allows updates from the machine names abacus. Click OK to return to the Set Access Permissions dialog box.

7. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

8. Double-click the cell under User/Group. The Select Users and Groups dialog box appears. Select SSL from the Authentication Method pull-down menu.

9. Double-click the cell under Host in the table. The Select Hosts and IP Addresses dialog box appears. Select “IP Address” from the drop-down menu, type the IP address of the host in the text box provided. For example, `123.45.6.78`, and then click Add. Click OK to return to the Set Access Permissions dialog box.

10. Click OK.

    The server saves the new ACI.

Note If other ACIs that allow access are set for this entry, then clients from other hosts or those not using SSL will still have access permissions. This ACI should be complemented with one or more ACIs that restrict the rights of other users and clients connecting from the host (unless you want to grant access to all users from the host).

Setting a Target Using Filtering

If you want to set access controls that allow access to a number of entries that are spread across the directory, you may want to use a filter to set the target. Keep in mind that because search filters do not directly name the object that you are managing access for, it is easy to unintentionally allow or deny access to the wrong objects, especially as your directory becomes more complex. Additionally, filters can make it difficult for you to troubleshoot access control problems within your directory.

The following procedure shows you how to grant user `bjensen` write access to the department number, home phone number, home postal address, JPEG photo, and manager attributes for all members of the accounting organization.

Before you can set these permissions, you must create the accounting branch point (`ou=accounting, o=airius.com`). You can create organizational unit branch points using the directory tab on the Directory Server Console. See Chapter 9, "Managing Directory Entries," for information. In addition, you can also add entries using the directory server gateway, or the Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* and the online help available through the gateway for information.

1. Make sure the directory server is running and that you have created the accounting branch point. For example, `ou=accounting, o=airius.com`.

2. Bind to the directory.

   You must enter the username and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the accounting entry in the navigation tree, for example, `ou=accounting, o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

The Set Access Permissions dialog box appears.

6. Click ACI Attributes.

   The Select Attributes dialog displays.

7. In the ACI Name field, enter a unique value such as `Accounting Filter`. In the future, this will allow you to see at a glance which of your access controls uses filtering to allow access to the accounting branch of the directory.

8. Select “is” from the Target Filter pull-down menu and enter the filter you want to use in the text box provided. For example, `ou=accounting`.

9. Select “is” from the Target Attribute(s) pull-down menu and type the following in the text box provided:

   ```
   departmentNumber || home* || jpegPhoto || Manager
   ```

10. Click OK to return to the Set Access Permissions dialog box.

11. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

12. Double-click the cell beneath User/Group in the table. The Select Users and Groups dialog box appears. Select Add User to List from the pull-down menu, type the user’s DN in the text box provided. For example, `uid=bjensen, o=airius.com`. Click Add. Click OK to return to the Set Access Permissions dialog box.

13. Click OK.

    The server saves the new ACI.

Allowing Users to Add or Remove Themselves From a Group

Many directories set ACIs that allow users to add or remove themselves from groups. This is useful, for example, for allowing users to add and remove themselves from mailing lists.

The following example allows anyone to add or delete themselves from the Jokes group.

Before you can set these permissions, you must create the jokes group (`cn=jokes, ou=Mail Server, o=airius.com`). You can create groups using the directory tab on the Directory Server Console. See Chapter 9, "Managing Directory Entries," for information. In addition, you can also add groups using the directory server gateway, or the Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* and the online help available through the gateway for information.

1. Make sure the directory server is running and that you have created the Jokes group. For example, `cn=jokes, ou=Mail Server, o=airius.com`.

2. Bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to all ACIs that have been set for the directory. See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.

3. On the Directory Server Console, select the Directory tab.

4. Right-click the jokes group entry in the navigation tree, for example, `cn=jokes, ou=Mail Server, o=airius.com`, and select Set Access Permissions from the pop-up menu.

   The Multi-value ACI Selector dialog box appears.

5. Click New.

   The Set Access Permissions dialog box appears.

6. Click ACI Attributes. The Select Attributes dialog displays. In the ACI Name field, enter a unique value such as `selfwrite jokes`. In the future, this will allow you to see at a glance which of your access controls allows users to add or remove themselves from the jokes group. Click OK to return to the Set Access Permissions dialog box.

7. Click the cell beneath Allow/Deny in the table and select Allow from the pull-down menu.

8. Double-click the cell beneath User/Group in the table. The Select Users and Groups dialog box appears. Select Add User to List from the pull-down menu, type ALL, and click Add. Click OK to return to the Set Access Permissions dialog box.

9. Double-click the cell beneath Rights in the table. The Select Rights dialog appears. Select the Selfwrite checkbox and click OK to return to the Set Access Permissions dialog box.

10. Click OK.

    The server saves the new ACI.

Setting Access Control Using LDIF Files

Access control information is stored in the ACI attribute of each directory entry. Because the access control information is stored in the directory, it can be managed using LDIF files. For a general description of the LDIF language, see Chapter 2, “LDAP Data Interchange Format.”

The ACI attribute is a special attribute; it is available for use on every entry in the directory, regardless of whether it is defined for the object class structure in use on the entry.

The following sections describe the ACI language syntax. You need to understand this syntax to use LDIF to manipulate access control permissions.

Tip LDIF ACI statements can be very complex. However, if you are setting access control for a large number of directory entries, using LDIF is the preferred method over using the GUI because of the time it can save. To familiarize yourself with LDIF ACI statements, however, you may want to use the server console to set the ACI and then click the View/Edit Syntax button on the Set Access Permissions dialog box. This shows you the correct LDIF syntax. If your operating system allows, you can even copy the LDIF and then paste it into your LDIF file.

The ACI Language Syntax

ACIs take the form

```
aci: (<target>)(version 3.0;aci"<name>";<permission><bind rule>;)
```

where

- `<target>` defines the object, attribute, or filter you are using to define what resource to control access to. The target can be a distinguished name, one or more attributes, and/or a single LDAP filter. For general information on targets, see “Targets” on page 99. For information on target syntax, see “Setting Targets Using LDIF” on page 137.
- `version 3.0` is a required string that identifies the ACI version.
- `aci "<name>"` is a name for the ACI. <name> can be any string that identifies the ACI. The ACI name is required.
- `<permission>` defines the actual access rights and whether they are to be allowed or denied. For general information on permissions, see “Permissions” on page 101. For information on the permission syntax, see “Setting Permissions Using LDIF” on page 140.
- `<bind rules>` identify the circumstances under which the directory login must occur in order for the ACI to take effect. For general information on bind rules, see “Bind Rules” on page 104. For information on bind rule syntax, see “Setting Bind Rules Using LDIF” on page 141.

The permission and bind rule portions of the ACI are set as a pair, and you can have multiple permission-bind rule pairs for each target. This allows you to efficiently set multiple access controls for a given target. For example:

```
<target>(<permission><bind rule>)(<permission><bind rule>)...
```

ACIs are order independent with regard to the `(<target>)` and `(<permission><bind rule>)` parts of the ACI.

The following is an example of a complete LDIF ACI:

```
aci: (target="ldap:///uid=bjensen,o=airius.com")(targetattr=*)
(version 3.0;acl "aci1";allow (write) userdn="ldap:///self";)
```

In this example, the ACI states that the user bjensen has rights to modify all attributes in her own directory entry.

The following sections describe the syntax of each portion of the ACI in more detail.

Setting Targets Using LDIF

Targeting indicates which directory resources to which an ACI applies. For general information on targeting, see “Targets” on page 99.

The general syntax for a target is:

```
aci: (<keyword> = "<expression>")
aci: (<keyword> != "<expression>")
```

where equal (=) indicates that the target is the resource specified in the `<expression>`, and not equal (!=) indicates the target is not the resource specified in the `<expression>`.

The quotation marks ("") around `<expression>` are required. What you use for `<expression>` is dependent upon the exact `<keyword>` that you supply.

The following table lists each keyword and the associated expressions.

Table 5.2 LDIF target keywords

Keyword	Valid Expressions	Wildcard allowed?
target	ldap:///<distinguished name>	yes
targetattr	<attribute>	yes
targetfilter	<LDAP search filter>	yes

The following sections further detail the target syntax for each keyword.

Using the target Keyword

To target an entry, use the target keyword. The target keyword can accept a value of the following format:

```
"ldap:///<distinguished name>"
```

This identifies the distinguished name of the entry to which the access control rule applies. For example:

```
(target = "ldap:///uid=bjensen, o=airius.com")
```

Note If the DN of the entry to which the access control rule applies contains a comma, you must escape the comma with a single backslash (\). For example:

```
(target = "ldap:///uid=lfuentes, o=Airius Bolivia\, S.A.")
```

You can also use a wildcard when targeting a distinguished name using the target keyword. The wildcard indicates that any character or string or substring is a match for the wildcard. Pattern matching is based on any other strings that have been specified with the wildcard. The following are legal examples of wildcard usage:

- `(target="ldap:///uid=*, o=airius.com")`

  Matches every distinguished name in the Airius tree that begins with `uid`.

- `(target="ldap:///uid=*Anderson, o=airius.com")`

  Matches every distinguished name who's `uid` ends with Anderson.

- `(target="ldap:///uid=C*A, o=airius.com")`

  Matches every distinguished name who's `uid` begins with C and ends with A.

- `(target="ldap:///uid=*, ou=*, o=airius.com")`

  Matches every entry in the Airius tree whose distinguished name contains the `uid` and `ou` attributes. Thus:

  ```
  uid=fchen, ou=Engineering, o=airius.com
  ```

  would match, but the following would not:

  ```
  uid=bjensen, o=airius.com
  ou=Engineering, o=airius.com
  ```

Note You cannot use wildcards in the suffix part of a distinguished name. That is, if your directory uses the suffixes

```
c=US
```

and

```
c=GB
```

then you can *not* use the following target to reference both suffixes:

```
(target="ldap:///o=airius, c=*")
```

For more information on using the target keyword, see “Targeting a Directory Entry” on page 99.

Using the targetattr Keyword

You can target a specific attribute by using the targetattr keyword. For example:

```
(targetattr = "cn")
```

The attribute identified on the targetattr keyword applies to the entry that the ACI is targeting, and to all that entry's child entries. That is, if you target the password attribute on the entry

```
uid=bjensen, ou=Marketing, o=airius.com
```

then only the password attribute on the bjensen entry is affected by the ACI. If however, you target the tree's branch point

```
ou=Marketing, o=airius.com
```

then all the entries beneath the branch point that can contain a password attribute are affected by the ACI.

For general information on targeting attributes, see “Targeting Attributes” on page 100.

Targeting Multiple Attributes

You can use the ACI's resource information to target multiple attributes by using the targetattr keyword along with the || symbol. For example, to target an entry's common name, surname, and uid attributes, use the following:

```
(targetattr = "cn || sn || uid")
```

Targeting Both an Entry and Attributes

By default, the entry targeted by an ACI containing a targetattr keyword is the entry on which the ACI is placed. That is, if you put the ACI

```
aci: (targetattr = "uid")(<access control rules>;)
```

on the `ou=Marketing, o=airius.com` entry, then the ACI applies to the entire Marketing subtree. However, you can also explicitly specify a target using the target keyword as follows:

```
aci: (target="ldap:///ou=Marketing, o=airius.com")
(targetattr = "uid")(<access control rules>;)
```

The order in which you specify the target, the targetattr and the access control rules is not important. Also, the entry you target using the target keyword must reside within the domain controlled by the entry on which the ACI is supplied. For more information on using the target keyword, see “Using the target Keyword” on page 137.

Using the targetfilter Keyword

You can target a group of entries and/or attributes using an LDAP search filter and the targetfilter keyword. For example:

```
(targetfilter = "<search operation>")
```

where <search operation> is a search operation in the standard LDAP search filter format. For information on the syntax of LDAP searches, see Chapter 8, “Finding Directory Entries.”

For example,

```
(targetfilter = "(|(ou=accounting)(ou=engineering))"
```

targets all entries in the accounting and engineering branches of the directory tree.

For more information on targeting using LDAP filters, see “Targeting Using LDAP Filters” on page 100.

Setting Permissions Using LDIF

Permissions indicate whether a specified right is allowed or denied. Whether the defined permission is allowed or denied is determined by whether the accompanying bind rule is evaluated to be true. For general information on permissions, see “Permissions” on page 101.

Permissions take the form

```
allow|deny (<rights>)
```

where

- `allow|deny` can be either the allow keyword or the deny keyword. Allow grants the user the permission and deny forbids the user the permission. For more information on allowing or denying access, see “Allowing or Denying Access” on page 102.
- `<rights>` is a list of 1 to 8 comma-separated keywords enclosed within parentheses. Valid keywords are read, write, add, delete, selfwrite, search, compare, proxy, or all. For a description of each access right, see “Assigning Rights” on page 103.

In the following example, read, search, and compare access is allowed provided the bind rule is evaluated to be true:

```
aci:  (target="ldap:///o=airius.com") (version 3.0;acl "acl2";
allow (read, search, compare) <bind rule>;)
```

Setting Bind Rules Using LDIF

Whether access is allowed or denied depends on whether an ACI’s bind rule is evaluated to be true. Bind rules use one of the two following patterns:

```
<keyword> = "<expression>";
```

```
<keyword> != "<expression>";
```

where equal (=) indicates that `<keyword>` and `<expression>` must match in order for the bind rule to be true, and not equal (!=) indicates that `<keyword>` and `<expression>` must not match in order for the bind rule to be true.

Note The `timeofday` keyword also supports the inequality expressions (<, <=, >, >=). This is the only keyword that supports these expressions.

The quotation marks (`""`) around `<expression>` are required as is the delimiting semicolon (;). What you use for `<expression>` depends on the exact `<keyword>` that you supply.

The following table lists each keyword and the associated expressions, and indicates whether wildcard characters are allowed.

Table 5.3 LDIF bind rule keywords

Keyword	Valid expressions	Wildcard allowed?
`userdn`	ldap:///<DN> ldap:///all ldap:///anyone ldap:///self ldap:///parent ldap:///<suffix>??sub?(filter)	yes, in DN only
`groupdn`	ldap:///<DN> || <DN>	no
`groupdnattr`	<attribute> ldap:///<DN>?<attribute>	no
`userdnattr`	<attribute>	no
`ip`	<IP address>	yes
`dns`	<DNS host name>	yes
`dayofweek`	sun mon tue wed thu fri sat sun	no
`timeofday`	0 - 2359	no
`authmethod`	none simple ssl sasl <authentication method>	no

The following sections further detail the bind rule syntax for each keyword.

Using the userdn Keyword

The `userdn` keyword requires one or more valid distinguished names in the format of `ldap:///<dn>[||ldap:///<dn>][...]`, the keyword `ldap:///self`, `ldap:///all`, or `ldap:///anyone`. For general information about user access, see "User Access" on page 106.

Note If a DN contains a comma, the comma must be preceded by a backslash (\) escape character.

The following are examples of the `userdn` syntax:

```
userdn = "ldap:///uid=*, o=airius.com";
```

The bind rule is evaluated to be true if the user binds to the directory using any distinguished name of the supplied pattern. For example, both of the following bind DNs would be evaluated as true:

```
uid=ssarette, o=airius.com
uid=bjensen, o=airius.com
```

whereas both of the following bind DNs would be evaluated as false:

```
cn=Babs Jensen, o=airius.com
uid=tjaz, ou=Accounting, o=airius.com
```

```
userdn = "ldap:///self";
```

The bind rule is evaluated to be true if the user is accessing the entry represented by the DN with which the user bound to the directory. That is, if the user has bound as `uid=ssarette, o=airius.com` and the user is attempting an operation on the `uid=ssarette` entry, then the bind rule is true.

```
userdn = "ldap:///all";
```

The bind rule is evaluated to be true for any valid bind DN. To be true, a valid distinguished name and password must have been presented by the user during the bind operation.

```
userdn = "ldap:///anyone";
```

The bind rule is evaluated to be true for anyone; use this keyword to provide anonymous access to your directory.

```
userdn = "ldap:///uid=bj, o=airius.com || ldap:///uid=kc, o=airius.com";
```

The bind rule is evaluated to be true if the client binds as either of the two supplied distinguished names.

```
userdn != "ldap:///uid=*, ou=Accounting, o=airius.com";
```

The bind rule is evaluated to be true if the client is not binding as a UID-based distinguished name in the accounting subtree. Keep in mind that this bind rule only makes sense if the targeted entry is above the accounting branch of the directory tree.

```
userdn = "ldap:///o=airius.com???(ou=engineering)(ou=sales)";
```

The bind rule is evaluated to be true if the client belongs to the engineering or sales subtree.

Using the groupdn Keyword

The `groupdn` keyword requires one or more valid distinguished names in the format of `ldap:///<dn> [|| ldap:///<dn> [|| ldap:///<dn>]...]`. The bind rule is evaluated to be true if the bind DN belongs to the named group. For general information about group access, see “Group Access” on page 107.

Note If a DN contains a comma, the comma must be escaped by a backslash (\).

The following are examples of the `groupdn` syntax:

```
groupdn = "ldap:///cn=Administrators, o=airius.com";
```

The bind rule is evaluated to be true if the bind DN belongs to the Administrators group.

```
groupdn = "ldap:///cn=Administrators, o=airius.com" ||
"ldap:///cn=Mail Administrators, o=airius.com";
```

The bind rule is evaluated to be true if the bind DN belongs to either the Administrators or the Mail Administrators group.

Using the userdnattr and groupdnattr Keywords

The `userdnattr` and `groupdnattr` keywords require an attribute that is expected to contain a full DN. The `userdnattr` keyword should reference an attribute that contains the DN of a single user whereas the `groupdnattr` keyword can reference an attribute that contains the DN of a user or group. For

example, the `manager` and `owner` attributes. For general information about user attribute access, see “User Access” on page 106. For general information about group access, see “Group Access” on page 107.

userdnattr example

The following is an example of the `userdnattr` syntax:

```
userdnattr = "manager";
```

The bind rule is evaluated to be true if the bind DN is the same as the value set for the manager attribute of the targeted entry. You might want to use this for allowing a user's manager to manage employees' password attributes.

groupdnattr examples

The following are examples of the `groupdnattr` syntax:

```
groupdnattr = "owner";
```

The bind rule is evaluated to be true if the bind DN is the same as the value set for the owner attribute of the targeted entry. In this example, the owner attribute is the “variable attribute”. You might want to use this for allowing a group to manage employees' status information. This is the same as the following example with two exceptions; the group you point to can be a dynamic group, and the DN of the group can be under any suffix in the database. However, the process by which the server evaluates this ACI syntax is more resource intensive. Use this syntax sparingly. See “Groups” in *Managing Servers with Netscape Console* for information on dynamic groups.

```
groupdnattr = "ldap:///o=airius.com?owner;
```

The bind rule is evaluated to be true if the bind DN is the same as the value set for the owner attribute of the targeted entry. You cannot use this syntax to target dynamic groups. Also, the DN of the group must be under the same suffix as the suffix specified in the `groupdnattr` portion of the aci. In this example, the DN of the group is under the `o=airius.com` suffix. However, you might want to use this syntax because the server processes it more quickly than the previous syntax example.

Using Inheritance with userdnattr and groupdnattr

Use inheritance to extend an ACI from a parent entry to include a number of its child entries. You add inheritance to an ACI using the `parent` parameter and the `userdnattr` or `groupdnattr` keywords.

The syntax for the `parent` parameter is as follows:

```
userdnattr = "parent[<inheritance_level>].<attribute>";
```

or

```
groupdnattr = "parent[<inheritance_level>.<attribute>";
```

Where `<inheritance_level>` is a comma separated list that indicates how many levels below the target you want to inherit the rule. You can set up to five levels of inheritance `[0,1,2,3,4]` below the targeted entry; where zero (0) indicates the targeted entry. The more levels of inheritance you specify, the bigger the negative impact on performance, so consider this option carefully before implementing it.

`<attribute>` is the attribute targeted by the `userdnattr` or `groupdnattr` keyword.

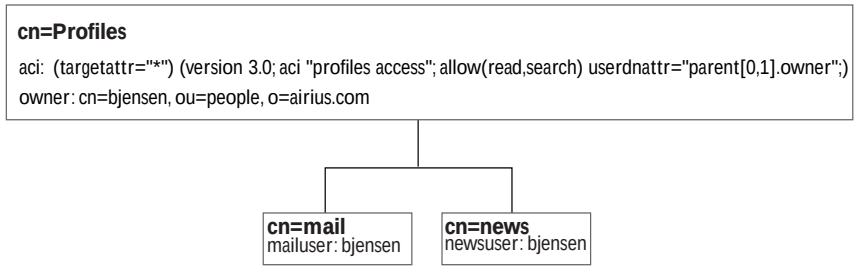
For example,

```
groupdnattr = "parent[0,1].manager";
```

userdnattr and groupdnattr Inheritance Example

The example in the following figure indicates that user `bjensen` is allowed to read and search the `cn=Profiles` entry as well as the first level of child entries which includes `cn=mail` and `cn=news` thus allowing her to search on her own mail and news IDs.

Figure 5.2 Using inheritance with the userdnattr keyword



In this example, if you did not use inheritance you would have to do one of the following to achieve the same result:

- Explicitly set read and search access on the `cn=Profiles`, `cn=mail`, and `cn=news` entries in the directory.
- Add the owner attribute with a value of `bjensen` to the `cn=mail` and `cn=news` entries and then add the following ACI to the `cn=mail` and `cn=news` entries.

  ```
  aci: (targetattr="*") (version 3.0; aci "profiles
  access"; allow (read,search) userdnattr="owner";)
  ```

Using the ip Keyword

The `ip` keyword requires an IP address. The LDIF syntax for setting a bind rule based on IP address is as follows:

```
ip = "<IP address>" or ip != "<IP address>"
```

Wildcards are allowed. For example:

```
ip = "12.345.6.*";
```

The bind rule is evaluated to be true if the client accessing the directory is located at the named IP address. This is useful for, for example, allowing certain kinds of directory access only from a specific subnet or machine.

Using the dns Keyword

The `dns` keyword requires a fully qualified DNS domain name. The LDIF syntax for setting a bind rule based on the DNS host name is as follows:

```
dns = "<DNS Hostname>" or dns != "<DNS Hostname>"
```

Wildcards are allowed. For example:

```
dns = "*.airius.com";
```

The bind rule is evaluated to be true if the client accessing the directory is located in the named domain. This is useful for, for example, allowing certain kinds of directory access only from a specific domain.

Using the timeofday Keyword

The `timeofday` keyword requires a time of day in the 24 hour clock (0 to 2359). Inequality expressions are allowed. The LDIF syntax for setting a bind rule based on the time of day is as follows:

```
timeofday <operator> "<time>"
```

where `<operator>` is equal to (=), not equal to (!=), greater than (>), greater than or equal to (>=), less than (<), or less than or equal to (<=).

Note The time on the server is used for the evaluation, and not the time on the client.

For example:

```
timeofday = "1200";
```

The bind rule is evaluated to be true if the client is accessing the directory at noon.

```
timeofday != "1200";
```

The bind rule is evaluated to be true if the client is accessing the directory at any time other than noon.

```
timeofday > "1200";
```

The bind rule is evaluated to be true if the client is accessing the directory at any time after noon.

```
timeofday < "1200";
```

The bind rule is evaluated to be true if the client is accessing the directory at any time before noon.

```
timeofday >= "1200";
```

The bind rule is evaluated to be true if the client is accessing the directory at noon or later.

```
timeofday <= "1200";
```

The bind rule is evaluated to be true if the client is accessing the directory at noon or earlier.

Using the dayofweek Keyword

The LDIF syntax for setting a bind rule based on the day of the week is as follows:

```
dayofweek = "<day>"
```

where <day> is one of the following: Sun, Mon, Tue, Wed, Thu, Fri, Sat. A list of values is allowed.

Note The day on the server is used for the evaluation, and not the day on the client.

For example:

```
dayofweek = "Sun, Mon, Tue";
```

The bind rule is evaluated to be true if the client is accessing the directory on Sunday, Monday, or Tuesday.

Using the authmethod Keyword

The LDIF syntax for setting a bind rule based on authentication method is as follows:

```
authmethod = "<authentication method>"
```

where <authentication method> is "none", "simple", "ssl", or "sasl <sasl mechanism>".

For example,

```
authmethod = "none";
```

Authentication is not checked during bind rule evaluation.

```
authmethod = "simple";
```

The bind rule is evaluated to be true if the client is accessing the directory using a username and password.

```
authmethod = "ssl";
```

The bind rule is evaluated to be true if the client authenticates to the directory using a certificate over ldaps. This is not evaluated to be true if the client authenticates using simple authentication (bind DN and password) over ldaps.

```
authmethod = "sasl kerberos";
```

The bind rule is evaluated to be true if the client is accessing the directory using the Kerberos SASL mechanism. The SASL mechanisms such as Kerberos are not provided as part of the directory server. For information on integrating SASL with the directory server, see the *Netscape Directory Server Programmer's Manual*.

Using Boolean Expressions in LDIF Bind Rules

Bind rules can be complex expressions that use the boolean expressions `AND`, `OR`, and `NOT`.

The general format for a Boolean bind rule is as follows:

```
<bind rule> [<boolean>][<bind rule>][<boolean>][<bind rule>]...;)
```

For example, the following bind rule will be evaluated as true if the bind DN is a member of either the administrator's group or the mail administrator's group, and if the client is running from within the Airius domain:

```
(groupdn = "ldap:///cn=administrators, o=airius.com" or groupdn =
"ldap:///cn=mail administrators, o=airius.com") and dns =
"*.airius.com";)
```

The trailing semicolon (;) is a required delimiter that must appear after the final bind rule.

ACI Usage Examples

The following examples describe how to set some of the more common directory permissions using the LDIF:

Defining Permissions for All Users

The following example allows all users at Airius read access to the directory. In the following examples, users must authenticate to the directory server to obtain access to the directory granted by this ACI:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target = "ldap:///o=airius.com") (targetattr=*)
(version 3.0; aci "all-read"; allow (read) userdn = "ldap:///all";)
```

The following example allows every user write privileges only to their own `userPassword` attribute:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (targetattr = "userPassword")
(version 3.0; aci "write-self"; allow (write) userdn = "ldap:///self";)
```

The following example allows every user to read every attribute except the `userPassword` attribute:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target = "ldap:///o=airius.com")(targetattr != "userPassword")
(version 3.0; aci "read-all-not-pw"; allow (read) userdn =
"ldap:///all";)
```

Defining Anonymous Access

Anonymous access allows users to gain access to the directory by providing no authentication information to the directory server. You grant anonymous access to an entry by specifying no client request attributes in the ACI.

The following example allows anonymous read and search access to every entry in the directory:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "anonymous-read-search"; allow (read, search)
userdn = "ldap:///anyone";)
```

The following example allows anonymous read or search access to the Marketing subtree:

```
dn: ou=Marketing, o=airius.com
objectClass: top
objectClass: organization
aci:: (target="ldap:///ou=Marketing, o=airius.com")(targetattr=*)
(version 3.0; aci "anonymous-search-mktg"; allow (read, search)
userdn = "ldap:///anyone";)
```

Defining Permissions for Individual Users

The following example allows a specific user read or search access to the Marketing subtree. The user will be required to authenticate before obtaining access to the directory:

```
dn: ou=Marketing, o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///ou=Marketing, o=airius.com")(targetattr=*)
(version 3.0; aci "bjensen-r-s-mktg"; allow (read, search)
userdn = "ldap:///uid=bjensen, ou=Marketing, o=airius.com";)
```

The following example allows a specific user write access to the `userPassword` or `telephoneNumber` attribute of his own entry:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (targetattr = "userpassword || telephonenumber")
(version 3.0; aci "ssarette-write-pw-phone"; allow (write)
userdn = "ldap:///uid=ssarette, o=airius.com";)
```

The following example allows either of the two users to write to the administrator's group. That is, both of these users can add and delete group members to the administrator's group or change attribute values for the group entry:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target = "ldap:///cn=administrators,o=airius.com")
(targetattr="member||uniquemember")(version 3.0; aci "write-admingrp";
allow (write)userdn = "ldap:///uid=ssarette, o=airius.com" ||
"ldap:///uid=bjensen, o=airius.com";)
```

The following example allows anyone to add or delete themselves from the Jokes group. This is useful, for example, for allowing users to add and remove themselves from mailing lists:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target = "ldap:///cn=jokes, ou=Mail Server, o=airius.com")
(targetattr="member")(version 3.0; aci "selfwrite-jokes"; allow
(selfwrite)userdn = "ldap:///all";)
```

The following example allows users to write access to the `description` and `jpegPhoto` attributes of their own entries:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (targetattr = "description || jpegPhoto")(version 3.0; aci
"write-jpeg-descrip-self"; allow (write) userdn = "ldap:///self";)
```

The following example allows the user to add and delete directory entries. However, because the ACI does not grant write permission, the user cannot modify the entry even if he created it himself.

```
dn: o=airius.com
objectClass: top
objectClass: organization
```

```
aci: (target = "ldap:///o=airius.com")(version 3.0; aci
"landrew-add-delete"; allow (add, delete) userdn = "ldap:///uid=landrew,
o=airius.com";)
```

The following example allows the user to write to his own child entries:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target = "ldap:///o=airius.com")(targetattr=*)(version 3.0; aci
"ssarette-parent"; allow (write) userdn = "ldap:///parent";)
```

Defining Permissions for a Group of Users

The following example allows a group of users called Administrators to write to the entire contents of the directory:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "Administrators-write"; allow (write) groupdn =
"ldap:///cn=Administrators, o=airius.com";)
```

The following example allows anyone who is in the Directory Administrator group and not in the Content Administrators group to write to the `uid` attribute of any entry in the directory:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (targetattr= "uid")
(version 3.0; aci "diradmins-write-UID"; allow (write)
groupdn = "ldap:///cn= Directory Administrators, o=airius.com" and
groupdn != "ldap:///cn = Content Administrators, o=airius.com";)
```

The following example grants a manager full access to his or her employee's entries. In order for this to work, the `manager` attribute on each of the employee's entries must be set to the manager's distinguished name:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "manager-write"; allow (all) userdnattr = "manager";)
```

The following example allows members of the HR group to add new entries to and delete entries from the people organizational unit:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///ou=people, o=airius.com")
(version 3.0; aci "HR-add-delete"; allow (add, delete) groupdn =
"ldap:///cn=HR, o=airius.com";)
```

The following example allows members of the Engineering admins group to modify the `homePhone`, `homePostalAddress`, and `manager` attributes of all entries in the Engineering business category. This example uses LDAP filtering to select all entries with businessCategory attributes set to Engineering:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (targetattr="departmentNumber || home* || Manager")
(targetfilter="businessCategory=Engineering")
(version 3.0; aci "eng-admins-write"; allow (write)
groupdn ="ldap:///cn=Engineering Admins, o=airius.com";)
```

Defining Permissions for a Specific Subtree

The following example allows a client to bind as the accounting organizational unit entry, and then have full access to all the entries beneath the accounting subtree:

```
dn: ou=Accounting, o=airius.com
objectClass: top
objectClass: organizationalUnit
userPassword: {crypt}dksfjaoewirsdkfj
aci: (target="ldap:///ou=Accounting, o=airius.com")(targetattr=*)
(version 3.0; aci "accounting-branch"; allow (all)
userdn = "ldap:///ou=Accounting, o=airius.com";)
```

The following example allows any member of the accounting subtree to add themselves to or delete themselves from the Bean Counters group:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///cn=Bean Counters, ou=Accounting, o=airius.com")
(targetattr="member")(version 3.0; aci "selfwrite-beancounters"; allow
(selfwrite)userdn = "ldap:///uid=*, ou=Accounting, o=airius.com";)
```

Defining Permissions for a Specific Location

The following example allows a client running on the machine `ldap.airius.com` full access to the directory:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "aci 2"; allow (all) dns = "ldap.airius.com";)
```

The following example allows anonymous searching of the directory as long as the client is running on the specified subnet:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "aci 2"; allow (read, search)
userdn = "ldap:///anyone" and ip = "2.3.1.*";)
```

Defining Permissions Based on the Day of Week or the Time of Day

The following example denies all access to everyone except the Directory Administrator's group from 1:00 am to 3:00 am (0100 to 0300) on Sunday, Tuesday, and Friday:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "aci 1"; allow (all)
groupdn = "ldap:///cn=Directory Administrators, o=airius.com";)
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "aci 2"; deny (all)
groupdn != "ldap:///cn=Directory Administrators, o=airius.com" and
dayofweek = "Sun, Tues, Fri" and
(timeofday >= "0100" and timeofday <= "0300");)
```

Defining Permissions Based on Authentication Method

The following example allows members of the Directory Administrators group full access to the directory provided that they bind to the directory using the Simple Authentication and Security Layer (SASL) protocol:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "aci 2"; allow (all)
groupdn = "ldap:///cn=Directory Administrators, o=airius.com" and
authmethod = "sasl kerberos";)
```

The following example allows read and write access from the machine named `mastermind.airius.com` if the bind is established using a Secure Sockets Layer (SSL) connection:

```
dn: o=airius.com
objectClass: top
objectClass: organization
aci: (target="ldap:///o=airius.com")(targetattr=*)
(version 3.0; aci "aci 1"; allow (read, write)
dns = "mastermind.airius.com" and authmethod = "ssl";)
```

Defining Permissions for DNs That Contain a Comma

DNs that contain commas require special treatement within your LDIF ACI statements. In the target and bind rule portions of the ACI statement, commas must be escaped by a single backslash (\). The following example illustrates this syntax:

```
dn: o=Airius Bolivia\, S.A.
objectClass: top
objectClass: organization
aci: (target="ldap:///o=Airius Bolivia\, S.A.")(targetattr=*)
(version 3.0; aci "aci 2"; allow (all)
groupdn = "ldap:///cn=Directory Administrators, o=Airius Bolivia\,
S.A.";)
```

Overview of Proxied Authorization

Using proxied authorization, you can enable an LDAP client application (such as the Administration Server in Netscape Delegated Administrator), to use a single thread with a single authentication to service multiple users making requests against the Directory Server. Instead of having to rebind for each request, the client application binds to the Directory Server and passes a proxy control (containing a *proxy DN*) as part of the LDAP operation the client application is attempting to perform.

The proxy DN is the DN of an entry that has access permissions to the target on which the client-application is attempting to perform an operation. The proxy DN must be contained within the control sent to the Directory Server. Once received, the Directory Server confirms that the client-application has proxy access rights to the entry affected by the operation, and then grants the rights defined for the proxy DN to the requesting client application.

The command-line utilities also include an argument (`-y`) that passes the proxy DN to the server allowing you to test the client-application controls you write. For more information, see:

- “Additional ldapsearch Parameters” on page 216 for information on using this parameter with `ldapsearch`.
- “Additional ldapmodify Parameters” on page 246 for information on using this parameter with `ldapmodify`.
- “Additional ldapdelete Parameters” on page 250 for information on using this parameter with `ldapdelete`.

This section contains information about:

Proxied Authorization ACI Syntax

Proxied authorization is an access right like read, write, and delete. You set the proxy right on the target and provide the bind DN of the client application in the ACI as follows:

```
allow(<access_rights>) <BindRule_Keyword>="<BindDN>"
```

where:

- *<access_rights>* must include `proxy` but may also include any other access rights, such as read or write, that you want to explicitly allow for the client application. For information on other rights supported by the Directory Server, see “Assigning Rights” on page 103.
- *<BindRule_Keyword>* is any of the allowable LDIF bind rule keywords, for example `userdn`. See Table 5.3 on page 142 for a list of allowable bind rule keywords.
- *<BindDN>* is the bind DN of the client application making a request against the directory. For example, “`uid=MoneyWizAcctSoftware, ou=InfrastructureProducts, o=Airius.com`”.

You do not specify a proxy DN in the ACI. The proxy DN must be included in the proxy control sent by the client application.

Proxied Authorization ACI Example

For this example, suppose:

- The client application’s bind DN is “`uid=MoneyWizAcctSoftware, ou=InfrastructureProducts, o=Airius.com`”.
- The targeted subtree to which the client application is requesting access is “`ou=Accounting, o=Airius.com`”.
- An Accounting Administrator with access permissions to the Accounting subtree exists in the directory.

Then in order for the client application to gain access to the Accounting subtree (using the same access permissions as the Accounting Administrator):

- The Accounting Administrator must have access permissions to the "`ou=Accounting, o=Airius.com`" subtree. For example, the following ACI granting all rights to the Accounting Administrator entry would suffice:

```
dn: ou=Accounting, o=Airius.com
aci:  (target="ldap:///ou=Accounting, o=Airius.com")(targetattr="*")
      (version 3.0; aci "allowAll-AcctAdmin"; allow(all)
      userdn="uid=AcctAdministrator,ou=Administrators,o=Airius.com")
```

- The following ACI granting proxy rights to the client application must exist in the directory:

```
dn: ou=Accounting, o=Airius.com
aci:  (target="ldap:///ou=Accounting, o=Airius.com")(targetattr="*")
      (version 3.0; aci "allowproxy-accountingsoftware";
      allow(proxy) userdn="uid=MoneyWizAcctSoftware,
      ou=InfrastructureProducts, o=Airius.com")
```

With this ACI in place, the MoneyWizAcctSoftware client application can bind to the directory and send a proxy control requesting that it be granted the same access rights as the entry it specifies in the proxy DN. (In the above example, the proxy DN sent in the control would be "`uid=AcctAdministrator, ou=Administrators, o=Airius.com`".)

Note You cannot use the Directory Manager's DN (Root DN) as a proxy DN. Attempts to do so will be unsuccessful. Also if the Directory Server receives more than one proxied authentication control, an error is returned to the client application and the bind attempt is unsuccessful.

Specifying Proxy Authorization Rights On a Target

You can set proxy authorization rights on a target in two ways. This section describes:

- "Setting Proxy Rights Using the Server Console" on page 161
- "Setting Proxy Rights Using the Command Line" on page 162

Setting Proxy Rights Using the Server Console

To set proxy rights on a target in the Directory, you create an ACI that specifies that the bind DN of the client application has proxy rights to the target.

To specify proxy rights on a target:

1. Make sure the directory server is running.
2. From the Directory Server Console, bind to the directory.

   You must enter the user name and password of a privileged directory user, such as the directory manager, who has access to ACIs that have been set for the directory. See “Binding to the Directory From Netscape Console” on page 27 for specific instructions.
3. Select the Directory tab.
4. Right-click the entry in the navigation tree for which you want to set proxy rights, and select Set Access Permissions from the pop-up menu (Figure 5.1).
5. Click New.

   The Set Access Permissions dialog box appears.

   The table lists the access control rules (ACRs) defined for this ACI. By default, the first ACR in the table denies access to everyone with the exception of the root DN (Directory Manager).
6. (Optional) Click ACI Attributes. The Select Attributes dialog appears.

   If you want to change the name of the ACI, type the new name in the ACI Name text box. The name can be any string you want to use to uniquely identify the ACI, for example, “Allow proxied authorization”. If you do not enter a name, the server uses “unknown”. Click OK to return to the Set Access Permissions dialog box.
7. Click the cell under Allow/Deny in the table and select Allow from the drop down menu.

8. Double-click the cell beneath User/Group in the table.

   The Select Users and Groups dialog box appears. Select Add User to List from the pull-down menu, type the client application's DN in the text box provided. For example, `uid=MoneyWizAcctSoftware, ou=InfrastructureProducts, o=airius.com`. Click Add and then click OK to return to the Set Access Permissions dialog box.

9. Double-click the cell under Rights in the table.

   The Select Rights dialog box appears. Select the checkbox next to Proxy.

   You can also explicitly set permissions for the client application by selecting other access rights in this dialog box. For information on other access rights you can set using this dialog box, see Table 5.1 on page 113.

10. Click OK when you are finished to return to the Set Access Permissions dialog box.

11. Click OK.

    The server saves the new ACI.

Setting Proxy Rights Using the Command Line

You can set proxy rights on an entry using LDIF from the command-line just as you would any other rights. See "Setting Access Control Using LDIF Files" on page 135 for information on using LDIF to set access control, and see "Proxied Authorization ACI Syntax" on page 159 for information on the ACI syntax to use when setting proxy authorization rights using LDIF.

Viewing the Access Control List for a Suffix

The Access Control List (ACL) is the complete list of all the ACIs under a single suffix in the directory. You can view this list by running the following `ldapsearch` command:

```
ldapsearch -h <host> -p <port> -b <baseDN> -D <rootDN> -w <rootPassword>
(aci=*) aci
```

See "Using the Command-Line Utilities" on page 32 for information on `ldapsearch`.

Chapter

6

Managing Password and Account Lockout Policies

Netscape Directory Server provides several security mechanisms to help you protect your directory data. Two of the most fundamental security mechanisms are password protection and account lockouts. Weak password and account lockout policies make your directory more vulnerable to break-ins.

This chapter contains the following sections:

Other security mechanisms are described in Chapter 5, "Managing Access Control," and Chapter 11, "Managing SSL."

Managing the Password Policy

A password policy is a set of rules that govern how passwords are used in a given system. The password policy mechanism provided by the Directory Server allows you to dictate such things as how short a password must be and whether users can reuse passwords. When users attempt to bind to the directory, the Directory Server compares the password with the value in the password attribute of the user's directory entry to make sure they match. The Directory Server also uses the rules defined by the password policy to ensure that the password is valid before allowing the user to bind to the directory.

Configuring the Password Policy

Although you are not required to do so, Netscape recommends that you set an account lockout policy in addition to a password policy. (See "Managing the Account Lockout Policy" on page 170 for more information.)

The password policy you configure applies to all users within the directory except for the Directory Manager (Root DN).

To set up or modify the password policy for your Directory Server:

1. On the Directory Server Console, select the Configuration tab and then the Database folder.

   The Database tabs appear in the right pane.

2. Select the Passwords tab in the right pane.

   This tab contains the password policy for the Directory Server.

3. You can specify that users must change their password the first time they log on by selecting the "User must change password after reset" checkbox.

   See "Password Change After Reset" on page 167 for more information about this parameter.

4. To specify that users can change their own passwords, select the “User May change password” checkbox.

   See “User-Defined Passwords” on page 167 for information on this parameter.

5. If you want, you can specify that users cannot change their password for a specific amount of time by entering the number of days in the "Allow Changes in X Day(s)" text box.

   See “Password Minimum Age” on page 169 for information on this parameter.

6. To configure the server to maintain a history list of passwords used by each user, select the “Keep Password History” checkbox.

   See “Password History” on page 169 for more information about password history lists.

7. If you configure the server to keep password histories, specify the number of passwords you want the server to keep for each user in the “Remember X Passwords” text box.

8. If you do not want user passwords to expire, select the “Password never Expires” radio button.

9. If you want users to have to change their passwords periodically, select the “Password Expires After X Days” radio button and then enter the number of days that a user password is valid.

   See “Password Expiration” on page 168 for information on this parameter.

10. If you have turned password expiration on, you need to specify how long before the password expires to send a warning to the user by entering the number of days in the “Send Warning X Days Before Password Expires” text box.

    See “Expiration Warning” on page 168 for information on this parameters.

11. If you want the server to check the syntax of a user password to make sure it meets the minimum requirements set by the password policy, select the "Check Password Syntax" checkbox.

    For more information about the type of syntax checking the server performs, see “Password Syntax Checking” on page 168.

12. If you turn password syntax checking on, you need to specify the minimum acceptable length in the "Password Minimum Length" text box.

    See “Password Length” on page 169 for more information on this parameter.

13. Specify what encryption method you want the server to use when storing passwords from the “Password Encryption” pull-down menu.

    Specific information about each encryption method is provided in “Password Storage Scheme” on page 170.

14. When you have finished making changes to the password policy, click Save.

Password Policy Parameters

This section describes the parameters you set to create a password policy for your server. The parameters are described in the following sections:

Password Change After Reset

The Directory Server password policy lets you decide whether users must change their passwords after the first login or after the password is reset by the administrator. Often the initial passwords set by the administrator follow some sort of convention, such as the user's initials, user ID, or the company name. Once the convention is discovered, it is usually the first value tried by a hacker trying to break in. In this case, it is a good idea to require users to change their passwords after such a change. If you configure this option for your password policy, users will be required to change their password even if user-defined passwords are disabled. (See "User-Defined Passwords" on page 167 for information.) If you choose to disallow users from changing their own passwords, administrator assigned passwords should not follow any obvious convention and should be difficult to discover.

By default, users do not need to change their passwords after reset.

User-Defined Passwords

You can set up your password policy to either allow or disallow users from changing their own passwords. A good password is the key to a strong password policy. Good passwords do not use trivial words—that is, any word that can be found in a dictionary, names of pets or children, birthdays, user IDs, or any other information about the user that can be easily discovered (or stored in the directory itself). Additionally, a good password should contain a combination of letters, numbers, and special characters. Often, however, users simply use passwords that are easy to remember. This is why some enterprises choose to set passwords for users that meet the criteria of a "good" password and disallow the users from changing the passwords. However, this approach requires a significant administrative effort. In addition, by providing passwords for users rather than letting them come up with passwords that are meaningful to them and therefore more easily remembered, you run the risk that they will write their passwords down somewhere where they can be discovered. By default, user-defined passwords are allowed.

Password Expiration

You can set your password policy so that users can use the same passwords indefinitely. Or, you can set your policy so that passwords expire after a given amount of time. In general, the longer a password is in use, the more likely it is to be discovered. On the other hand, if passwords expire too often, users may have trouble remembering them and resort to writing their passwords down. A common policy is to have passwords expire every 30 to 90 days.

The server remembers the password expiration even if you turn the password expiration feature off. This means that if you turn the password expiration option back on, passwords will be valid only for the duration you had set before you last disabled the feature. For example, suppose you set up passwords to expire every 90 days and then decided to disable password expiration. When you decide to re-enable password expiration, the default password expiration duration is 90 days because that is what you had it set to before you disabled the feature. By default, user passwords never expire.

Expiration Warning

If you choose to set your password policy so that user passwords expire after a given number of days, it is a good idea to send users a warning before their passwords expire. You can set your policy so that users are sent a warning 1 to 24,855 days before their passwords are due to expire. The Directory Server displays the warning when the user binds to the server. If password expiration is turned on, by default, a warning will be sent (via LDAP message) to the user one day before the user's password expires, provided the user's client application supports this feature. Both Netscape Directory Express and the Netscape Directory Server Gateway provide this functionality.

Password Syntax Checking

The password policy establishes some syntax guidelines for password strings, such as the minimum password length guideline. The password syntax-checking mechanism ensures that the password strings conform to the password syntax guidelines established by the password policy. Additionally, the password syntax-checking mechanism also ensures that the password is not a "trivial" word. A trivial word is any value stored in the `uid`, `cn`, `sn`, `givenName`, ou, or `mail` attribute of the user's entry. By default, password syntax checking is turned off.

Password Length

The Directory Server allows you to specify a minimum length for user passwords. In general, shorter passwords are easier to crack. You can require passwords that are from 2 to 512 characters. A good length for passwords is 8 characters. This is long enough to be difficult to crack, but short enough that users can remember the password without writing it down. By default, no minimum password length is set.

Password Minimum Age

You can configure the Directory Server to disallow users from changing their passwords for a given period of time. You can use this feature in conjunction with the Password History parameter to discourage users from reusing old passwords. Setting the password minimum age parameter to 2 days, for example, prevents a user from repeatedly changing her password during a single session in order to cycle through the password history and reuse an old password once it is removed from the history list. You can specify any number from 0 to 24,855 days. A value of zero (0) indicates that the user can change the password immediately.

Password History

You can set up the Directory Server to store from 2 to 24 passwords in history, or, you can disable password history, thus allowing users to reuse passwords.

If you set up your password policy to enable password history, the directory stores a specific number of old passwords. If a user attempts to reuse one of the passwords the Directory Server has stored, the password will be rejected. This feature prevents users from reusing a couple of passwords that are easy to remember.

The passwords remain in history even if you turn the history feature off. This means that if you turn the password history option back on, users will not be able to reuse the passwords that were in history before you disabled password history. By default, the server does not maintain a password history.

Password Storage Scheme

The password storage scheme specifies the type of encryption used to store Directory Server passwords within the directory. You can specify

- no encryption
- SHA (Secure Hash Algorithm). Default.
- crypt (Unix crypt algorithm)

Although passwords stored in the directory can be protected through the use of access control information (ACI) instructions, it is still not a good idea to store cleartext passwords in the directory. SHA is the most secure of the choices and the one Netscape recommends. The crypt algorithm provides compatibility with Unix passwords.

Managing the Account Lockout Policy

The lockout policy works in conjunction with the password policy to provide further security. The account lockout feature protects against hackers who try to break into the directory by repeatedly trying to guess a user's password.

Configuring the Account Lockout Policy

Although you are not required to do so, Netscape recommends that you set a password policy in addition to an account lockout policy. (See "Managing the Password Policy" on page 164 for more information.)

To set up or modify the account lockout policy for your Directory Server:

1. On the Directory Server Console, select the Configuration tab and then the Database folder.

   The Database tabs appear in the right pane.

2. Select the Account Lockout tab in the right pane.

3. To enable account lockout, select the “Accounts may be locked out” checkbox.

   For specific information on this parameter, see “Account Lockout” on page 172. Clear this checkbox if you do not want to set an account lockout policy for the server.

4. Enter the maximum number of allowed bind failures in the “Lockout Account After X Login Failures” text box. The server locks out users who exceed the limit you specify here.

5. Enter the number of minutes you want the server to wait before resetting the bind failure counter to `0` in the “Reset Failure Counter After X Minutes” text box.

   For specific information about this parameter, see “Password Failure Counter Reset” on page 172.

6. Set the period of time you want users to be locked out of the directory.

   You can set it up so that users are locked out until their passwords are reset by the administrator by selecting the Lockout Forever radio button. Or, you can set it up so that users are locked out from 1 to 35,791,394 minutes by selecting the Lockout Duration radio button and entering the time in the available text box. For more information on this parameter, see “Lockout Duration” on page 172.

7. When you have finished making changes to the account lockout policy, click Save.

Account Lockout Policy Parameters

This section describes the parameters you set to create an account lockout policy for your server. The following parameters are described:

- “Account Lockout” on page 172
- “Password Failure Counter Reset” on page 172
- “Lockout Duration” on page 172

Account Lockout

You can choose not to lock users out regardless of the number of failed bind attempts. Alternatively, you can set up your policy to lock out users after 1 to 32,767 failed bind attempts.

Often when hackers are trying to break into a system, they will repeatedly try to guess a user's password. To prevent this type of attack, you can set up your password policy so that a specific user will be locked out of the directory after a given number of failed attempts to bind to the directory.

The server remembers the account lockout policy even if you turn the lockout feature off. This means that if you turn the account lockout option back on, users will by default be locked out after the number of failed bind attempts you had set before you last disabled account lockout. For example, suppose you set up account lockout so that users are locked out after 7 failed bind attempts and then decided to disable account lockout. If you decide to re-enable account lockout, the default number of failed bind attempts after which users are locked out is 7 because that is what you had it set to before you disabled the feature.

Password Failure Counter Reset

The Directory Server requires that you specify that the password failure counter be reset every 1 to 35,791,394 minutes.

Each time an invalid password is sent from the user's account, the password failure counter is incremented. When the counter reaches the number of tries specified by the account lockout parameter, the user will be locked out of the directory for the amount of time specified by the lockout duration parameter. Because the counter's purpose is to gauge when a hacker is trying to gain access to the system, the counter must continue for a period long enough to detect a hacker. However, if the counter was to increment indefinitely over days and weeks, valid users would probably be locked out inadvertently at some point.

Lockout Duration

If you enable account lockout, you can also set the period of time users will be locked out of the directory. You can set it up to lock users out until their passwords are reset by the administrator, or you can set it up so that users are locked out from 1 to 35,791,394 minutes.

Setting User Passwords

Because user passwords are stored in the directory, you can use whatever LDAP operation you normally use to update the directory to set or reset the user passwords.

For information on creating and modifying directory entries, see Chapter 9, “Managing Directory Entries.”

You can also use the Users and Groups area of the Administration Server or the Directory Server gateway to set or reset user passwords. For information on how to use the Users and Groups area, see the online help that is available through the Administration Server. For information on how to use the gateway to create or modify directory entries, see the online help that is available through the gateway.

Chapter

7

Managing Indexes

Netscape Directory Server uses index files to aid in searching the directory. Indexes greatly improve the performance of searches in the directory databases, but they do so at the cost of slower database modification and creation operations. Indexes also cost more in terms of system resources, especially disk space.

This chapter discusses how you can use indexes in the following sections:

The Searching Algorithm

The directory server uses the following process when performing a search:

1. An LDAP client such as Netscape Communicator or the directory server gateway sends a search filter to the directory server.

2. The directory server examines the incoming request to make sure that the specified base DN matches a suffix assigned to the local database.

   - If they do match, the directory server processes the request.

   - If they do not match, the directory server returns an error to the client indicating that the suffix does not match. If the `referral` parameter is set, the directory server also returns the LDAP URL where the client can attempt to pursue the request.

3. If the search filter criteria may be satisfied by a single index, then the server reads that index to generate a list of potential, or candidate, matches.

   If an appropriate index is not found, the directory server generates a candidate list that includes all entries in the database. (The directory server will also do this if the All IDs token is set for the index key that the server is using. For information on All IDs, see “Managing All IDs Threshold” on page 197.)

   If a search filter containing multiple criteria is used, the directory server consults multiple indexes and then combines the resulting lists of candidate entries.

4. The candidate matches are taken from the index files in the form of a series of entry ID numbers.

5. The directory server uses the returned entry ID numbers to read the corresponding entries from the `id2entry.db2` file. The directory server then examines each of the candidate entries to see if any match the search criteria. Matching entries are returned to the client as each is found.

The directory server continues until it either has examined all candidate entries, or until it reaches the limit set on one of the following parameters:

- Size Limit
- Time Limit
- Look Through Limit

The client may further restrict the number of returned entries by specifying lower values for the Size Limit and the Time Limit parameters on the search request.

Types of Indexes

To improve searching efficiency, the directory server can maintain six index types:

pres—Presence Index. Improves searches for entries that contain the indexed attribute.

eq—Equality Index. Improves searches for entries that contain an attribute that is set to a specific value.

approx—Approximate Index. Used only for string values such as `commonName` or `givenName`. Improves phonetic, or "sounds-like," searching.

sub—Substring Index. Improves searches for entries that contain a specified substring.

matching rule—International Index. Allows for searches that return entries sorted according to a specified collation order.

browse—Browsing Index. This index is used to enhance the browsing speed of the Directory Server Console.

Note The server stores indexes in files. The names of the files are based on the indexed attribute, not the type of index contained in the file. Each index file may contain multiple types of indexes if multiple indexes are maintained for the specific attribute. For example, all the indexes maintained for the common name attribute are contained in the `cn.db2` file.

Presence Index

The presence (pres) index is the simplest of the indexes. It lets you efficiently search the directory for entries that contain a specific attribute. This is useful if, for example, you want to examine any entries that have access control information associated with them. Generating an `aci.db2` file that includes a presence index lets you efficiently perform the search for `ACI=*` in order to generate the Access Control List for the server.

Equality Index

The equality (eq) index allows you to efficiently search for entries containing a specific attribute value. For example, an equality index on the `cn` attribute allows a user to efficiently perform the search for `cn=Babs Jensen`.

Approximate Index

The approximate (approx) index allows efficient approximate or "sounds-like" searches. For example, an entry may include the attribute value `cn=Robert E Lee`. An approximate search would return this value for searches against `cn~=Robert Lee`, `cn~=Robert`, or `cn~=Lee`. Similarly, a search against `l~=San Fransico` (note the misspelling) would return entries including `l=San Francisco`.

The directory server uses a variation of the metaphone phonetic algorithm to perform this type of searching. Each value is treated as a sequence of words, and a phonetic code is generated for each word.

Values entered on an approximate search are similarly translated into a sequence of phonetic codes. An entry is considered to match a query if both of the following are true:

- All of the query string codes are present in the codes generated in the entry string.
- All of the query string codes are specified in the same order as the entry string codes.

For example:

Name in the directory (Phonetic code)	Query string (Phonetic code)	Match comments
Alice B Sarette (ALS B SRT)	Alice Sarette (ALS SRT)	Matches. Codes are specified in the correct order.
	Alice Sarrette (ALS SRT)	Matches. Codes are specified in the correct order despite the misspelling of Sarette.
	Surette (SRT)	Matches. The generated code exists in the original name despite the misspelling of Sarette.
	Bertha Sarette (BR0 SRT)	No match. The code BR0 does not exist in the original name.
	Sarette, Alice (SRT ALS)	No match. The codes are not specified in the correct order.

Substring Index

The substring (sub) index is a costly index to maintain, but it allows efficient searching against substrings within entries.

For example, searches of the form:

```
cn=*derson
```

would match the common names containing strings such as

```
Bill Anderson
Jill Anderson
Steve Sanderson
```

Similarly, the search for

```
telephonenumber= *555*
```

would return all the entries in your directory with telephone numbers that contain 555.

Note Substring indexes are limited to a minimum of two characters for each entry.

International Index

The international index speeds up searches for information in international directories. The process for creating an international index is similar to the process for creating regular indexes, except that you apply a matching rule by associating a locale (OID) with the attributes to be indexed. For a listing of supported locales and their associated OIDs, see Table B.1 on page 495. If you want to configure the directory server to accept additional matching rules, see the *Netscape Directory Server Programmer's Manual*.

For more information on creating an international index, see "Creating Indexes" on page 183.

Browsing Index

The browsing index, or virtual list view index, speeds up the display of entries in the Directory Server Console. This index is particularly useful if you have a branch that contains hundreds of entries, for example, the `ou=people` branch. You can create a browsing index on any branchpoint in the directory tree to improve display performance. You do this through the Directory Server Console or by using the `vlvindex` command-line tool.

For information on creating and removing browsing indexes, see "Using Browsing Indexes" on page 195.

The Cost of Indexing

Although indexes greatly improve the speed of searches within the directory server database, it helps to be aware of the costs related to indexing as described in the following sections:

- "Slower Database Modification and Creation Times" on page 181
- "Higher System Resource Use" on page 182

Slower Database Modification and Creation Times

The more indexes you maintain, the longer it takes the directory server to update the database. This is especially true for substring indexes that cause the directory server to generate multiple index file entries every time an attribute value is created or changed. Remember that for substring indexes, the number of index entries created is proportional to the length of the string being indexed.

Consider the procedure for creating a specific attribute:

1. The directory server receives an `add` or `modify` operation.
2. The directory server examines the indexing parameters to determine whether an index is maintained for the attribute values.
3. If the created attribute values are indexed, then the directory server generates the new index entries.
4. Once the server completes the indexing, the actual attribute values are created according to the client request.

For example, suppose the directory server is asked to add the entry

```
dn: cn=Bill Pumice, ou=People, o=airius.com
objectclass: top
objectClass: person
objectClass: orgperson
objectClass: inetorgperson
cn: Bill Pumice
cn: Bill
sn: Pumice
ou: Manufacturing
ou: people
telephonenumber: 408 555 8834
description: Manufacturing lead for the Z238 line.
```

Further suppose that the directory server is maintaining these indexes:

- Equality, approximate, and substring indexes for common name and surname attributes
- Equality and substring indexes for the telephone number attribute

- Substring indexes for the description attribute

Then to add this entry to the directory, the directory server must perform these steps:

1. Create the common name equality index entry for “Bill” and “Bill Pumice”.
2. Create the appropriate common name approximate index entries for “Bill” and “Bill Pumice”.
3. Create the appropriate common name substring index entries for “Bill” and “Bill Pumice”.
4. Create the surname equality index entry for “Pumice”.
5. Create the appropriate surname approximate index entry for “Pumice”.
6. Create the appropriate surname substring index entries for “Pumice”.
7. Create the telephonenumber equality index entry for “408 555 8834”.
8. Create the appropriate telephonenumber substring index entries for “408 555 8834”.
9. Create the appropriate description substring index entries for “Manufacturing lead for the Z238 line of foobar widgets.” A large number of substring entries are generated for this string.

Higher System Resource Use

One other cost to maintaining index files is the increased system resources they require. Although the impact to your system depends on how large a database you use and how many attributes exist within your database, consider these facts:

- Index files use disk space—The more attributes you index, the more files will be created. Also, creating approximate and substring indexes for attributes that contain long and complex strings can cause these files to quickly grow large.

- Index files use memory—To run more efficiently, the directory server tries to place as many index files in memory as possible. You can control the amount of memory allowed per open index file using the “Maximum Cache Size” parameter. Even so, a large number of index files will require more memory.
- Managing index files uses CPU cycles—Although index files will save you CPU cycles during searches, maintaining indexes that are not frequently used can actually waste CPU cycles because of the need to create and manage unnecessary indexes. This is especially true for substring and approximate indexes that require the parsing of long, complex strings.

Creating Indexes

You create presence, equality, approximate, substring, and international indexes for specific attributes by using the “Attribute to be Indexed” parameter. (For instructions on managing browsing indexes, see “Using Browsing Indexes” on page 195.) Before you create new indexes, balance the benefits of maintaining indexes against the cost of indexing as described in “The Cost of Indexing” on page 180. Consider that:

- Approximate indexes make little sense for attributes commonly containing numbers, such as telephone numbers.
- Substring indexes make no sense for attributes that do not contain readable strings, such as attributes intended to contain a photograph or password attributes that contain encrypted data.
- Maintaining indexes for attributes that will not be commonly used in a search increases overhead without improving searching performance.
- Attributes that are not indexed can still be specified in search filters, although the search performance may be degraded significantly, depending on the type of search.

System and Default Indexes

When you install the directory server, some indexes are created by default. In addition, some default indexes are system indexes that you cannot remove. All of these indexes help to improve directory server performance.

Directory-enabled Netscape servers also maintain a standard set of indexes in the directory. If you are not using the server for which these indexes are maintained, you can either leave them alone (they do not require any resources if you are not using the corresponding server), or you can delete them from your server. For more information, see “Removing Indexes” on page 193.

System Indexes

The system indexes, in Table 7.1, cannot be removed.

Table 7.1 System indexes

Attribute	Eq	Pres	Purpose
aci		X	Allows the directory server to quickly obtain the access control information maintained in the database.
changeNumber	X		Used to improve replication performance.
copiedFrom		X	Used to improve replication performance.
dnComp	X		Used to help accelerate subtree searches in the directory.
objectClass	X		Used to help accelerate subtree searches in the directory.
entryDN	X		Speeds up entry retrieval based on DN searches.
parentID	X		Enhances directory performance during one-level searches.
numSubordinates		X	Used by the Directory Server Console to enhance display performance on the Directory tab.

Default Indexes

Default indexes are created for your server during installation. Although you can remove the default indexes, you should ensure that no server plug-ins or other Netscape servers in your enterprise require the index before you remove it. For information on removing indexes, see “Removing Indexes” on page 193.

Table 7.2 contains a list of default indexes and their purpose.

Table 7.2 Default indexes

Attribute	Eq	Pres	Sub	Purpose
cn	X	X	X	Improves the performance of the most common types of user directory searches.
givenName	X	X	X	Improves the performance of the most common types of user directory searches.
mail	X	X	X	Improves the performance of the most common types of user directory searches.
mailAlternateAddress	X			Improves NT Synchronization Service performance, and has no impact on your directory server's performance if you are not using the NT Synchronization Service.
mailHost	X			Used by the Netscape Messaging Server.
member	X			Improves Netscape server performance. This index is also used by the referential integrity plug-in. See “Managing the Referential Integrity Plug-in” on page 87 for more information.
nsCalXItemId	X	X	X	Used by the Netscape Calendar Server.
nsLIProfileName	X			Used by Netscape Communicator.

Table 7.2 Default indexes (Continued)

Attribute	Eq	Pres	Sub	Purpose
ntGroupDomainId	X	X	X	Improves NT Synchronization Service performance, and has no impact on your directory server's performance if you are not using the NT Synchronization Service.
ntUserDomainId	X	X	X	Improves NT Synchronization Service performance, and has no impact on your directory server's performance if you are not using the NT Synchronization Service.
owner	X			Improves Netscape server performance. This index is also used by the referential integrity plug-in. See "Managing the Referential Integrity Plug-in" on page 87 for more information.
pipstatus	X			Used by the Netscape Calendar Server.
pipuid	X	X	X	Used by the Netscape Calendar Server.
seeAlso	X			Improves Netscape server performance. This index is also used by the referential integrity plug-in. See "Managing the Referential Integrity Plug-in" on page 87 for more information.
sn	X	X	X	Improves the performance of the most common types of user directory searches.
telephoneNumber	X	X	X	Improves the performance of the most common types of user directory searches.

Table 7.2 Default indexes (Continued)

Attribute	Eq	Pres	Sub	Purpose
uid	X			Improves Netscape server performance.
uniquemember	X			Improves Netscape server performance. This index is also used by the referential integrity plug-in. See “Managing the Referential Integrity Plug-in” on page 87 for more information.

Standard Index Files

Because of the need to maintain default indexes and other internal indexing mechanisms, the directory server maintains several standard index files:

- `id2entry.db2`—contains the actual directory database entries. All other database files can be recreated from this one, if necessary.
- `id2children.db2`—restricts the scope of one-level searches, that is, searches that examine an entry's immediate children.
- `dn.db2`—controls the scope of subtree searches; that is, searches that examine an entry and all the entries in the subtree beneath it.
- `dn2id.db2`—begins all searches efficiently by mapping an entry's distinguished name to its ID number.

Creating Indexes From the Server Console

During the process of creating an index from the server console, the server is placed in read-only mode. While the server is in read-only mode, you cannot make any configuration changes or modify the contents of the directory.

Follow these steps to create presence, equality, approximate, substring, and international indexes for specific attributes from the Directory Server Console. (For instructions on managing browsing indexes, see “Using Browsing Indexes” on page 195.)

1. On the Directory Server Console, select the Configuration tab and then the Database icon.

2. Select the Indexes tab in the right pane.

   The System Indexes table contains indexes that you cannot delete. The Additional Indexes table contains the default indexes and any indexes you create. See “System and Default Indexes” on page 184 for more information.

3. If the attribute you want to index is listed in the Additional Indexes table on the tab, skip to Step 4, otherwise, click Add Attribute.

   A dialog box appears containing a list of all of the available attributes in the server schema. Select the attribute you want to index and click OK. The server adds the attribute to the Additional Indexes table.

4. Select the checkbox for each type of index you want maintained for the attribute.

5. If you want to create an index for a language other than English, enter the OID for the collation order you want to use in the Matching Rules field.

   You can index the attribute using multiple languages by listing multiple OIDs separated by commas (but no whitespace). For a list of languages and their associated OIDs, see Table B.1 on page 495.

6. Click Save.

   A dialog box appears displaying the status of the index creation and warning you that the server is in read-only mode while the indexes are created. While the server is in read-only mode, you cannot make any configuration changes or modify the contents of the directory. The server will display a dialog box informing you when the indexes are created and the server is no longer in read-only mode.

   The new index is immediately active for any new data that you add and any existing data in your directory. You do not have to restart your server.

Creating Indexes From the Command-Line

Creating indexes for attributes using the command line is a two-step process. First, you need to add the corresponding index description to `slapd.ldbm.conf`.

If you have existing data in your database that was not previously indexed, you also need to complete one of the following additional tasks, in order for the server to create the indexes you specify in `slapd.ldbm.conf`:

- Export and reimport the database using LDIF, (for instructions, see “Managing Databases Using LDIF” on page 70)
- Run the `db2index` command-line tool (for information, see “Creating Indexes Using db2index” on page 191).

Adding Index Descriptions to slapd.ldbm.conf

To add presence, equality, approximate, substring, or international index descriptions to `slapd.ldbm.conf` do the following:

1. Stop the server. See “Starting and Stopping the Directory Server” on page 29 for information.
2. Open `slapd.ldbm.conf`. For a description of where the configuration files reside in your directory server installation, see “Directory Server Configuration Files” on page 36.
3. Add an `index` parameter of the form:

   `index <`*`attribute`*`> [<`*`list of indexes`*`>][<`*`list of OIDs`*`>]`

   where `<`*`attribute`*`>` is the attribute to be indexed, `<`*`list of indexes`*`>` can include `pres`, `eq`, `sub`, and/or `approx`, and `<`*`list of OIDs`*`>` is a list of international collation OIDs. If you omit `<`*`list of indexes`*`>` and instead you specify just `<`*`list of OIDs`*`>` you must denote the null list of indexes with double quotation marks (“”).

You should always use the attribute's primary name (not the attribute's alias) when you create the index. The attribute's primary name is the first name listed for the attribute in the schema. See the *Netscape Directory Server Schema Reference Guide* for more information about attribute names.

For a definition of the index keywords, refer to "Types of Indexes" on page 177. For a list of available OIDs, see Table B.1 on page 495.

For example:

```
index cn,telephoneNumber pres,eq,sub
index cn "" 2.16.840.1.113730.3.3.2.15.1
index cn eq,sub 2.16.840.1.113730.3.3.2.15.1
```

Specifying an index parameter and attribute name with no indexes causes all indexes (except international) to be maintained for the specified attribute. For example:

```
index cn
```

causes all indexes to be maintained for the common name attribute. Use the keyword `none` to specify that no indexes are to be maintained for the attribute. For example:

```
index cn none
```

Use the keyword `default` to specify the list of all attributes not otherwise identified in an `index` parameter. For example:

```
index default none
```

4. Start the server.

5. When you finish editing the `slapd.ldbm.conf` file and start the server, the server will index any new entries. If you have existing data in your database that was not previously indexed, you also need to complete one of the following additional tasks, in order for the server to create the indexes you specify in `slapd.ldbm.conf`:

   - Export and reimport the database using LDIF, (for instructions, see "Managing Databases Using LDIF" on page 70)
   - Run `db2index` (for instructions, see "Creating Indexes Using db2index" on page 191).

Creating Indexes Using db2index

You can create new presence, equality, approximate, substring, and international indexes from the command line using the `slapd` (Windows NT) or `ns-slapd` (Unix) command-line utility with the `db2index` keyword.

For information on where you can find the command-line utilities in your directory server installation, see “Finding the Command-Line Utilities” on page 33.

Before you can create an index using `db2index`, you need to add the index description to `slapd.ldbm.conf`. See “Adding Index Descriptions to slapd.ldbm.conf” on page 189 for more information.

To create an index from the command line:

1. You can run `db2index` with the server shut down or running.

   Stopping the directory server and then running `db2index` is quicker. If you choose to do this, skip to the next step. However, if you want to continue to read information from the directory during the indexing process, you need to place the server in read-only mode. See “Placing a Database in Read-Only Mode” on page 85 for information.

2. From the command line, change to `<`*`NSHOME`*`>/bin/slapd/server`, where *`<NSHOME>`* is the directory where you installed the directory server.

3. Run the `slapd` (Windows NT) or `ns-slapd` (Unix) command-line utility as follows. Parameters in brackets [ ] are optional.

   Windows NT:

   ```
   slapd db2index -f <slapd.conf> [-d <debug_level>]
   [-n backend-number]
   -t attributeName[:indexTypes[:matchingRules]]
   ```

   Unix:

   ```
   ns-slapd db2index -f <slapd.conf> [-d <debug_level>]
   [-n backend-number]
   -t attributeName[:indexTypes[:matchingRules]]
   ```

where

- <*attributeName*> is the name of the attribute to be indexed.
- <*indexTypes*> is a comma-separated list of indexes to be created for the attribute.
- <*matchingRules*> is an optional comma-separated list of the OIDs for the languages in which you want the attribute to be indexed. This option is used to create international indexes. For information on supported locales and collation order OIDs, see Table B.1.

4. If you kept the server running during the process, stop the directory server and take the database out of read-only mode.

5. Start the server.

Parameters Used for Index Creation

The following `db2index` parameters are used to complete index creation:

-d. Optional. Specifies the debug level to use during index creation. Debug levels are defined in “Log Level” on page 444.

-f. Specifies the `slapd.conf` configuration file to use for the index creation process. Use the full path to the `slapd.conf` file with this argument. For information on where to find directory server configuration files, see “Directory Server Configuration Files” on page 36.

-n. Reserved for use by the directory server.

-t. Specifies the attribute to be indexed as well as the types of indexes to create and matching rules to apply (if any). If you want to specify a matching rule, you must specify an index type.

db2index Example

Windows NT:

```
slapd db2index
-f c:\Netscape\SuiteSpot4\slapd-dirserver\config\slapd.conf
-t uid:eq,pres,sub:2.16.840.1.113730.3.3.2.11.1
```

Unix:

```
ns-slapd db2index -f /usr/ns-home/slapd-dirserver/config/slapd.conf
-t uid:eq,pres,sub:2.16.840.1.113730.3.3.2.11.1
```

Removing Indexes

You can delete all presence, equality, approximate, substring, and international indexes that you have created and all default indexes that are not also system indexes. This process is outlined in the following sections:

- “Removing Indexes Using the Server Console” on page 193
- “Removing Standard Indexes Using the Command Line” on page 194

For instructions on removing a browsing index, see “Using Browsing Indexes” on page 195.

You cannot delete system indexes. These indexes are listed in Table 7.1 on page 184. For more information on these indexes, see “System and Default Indexes” on page 184.

Removing Indexes Using the Server Console

You can remove any indexes you have created, indexes used by other Netscape servers (such as Messaging or Calendar server), and default indexes that are not also system indexes using the Directory Server Console. To remove indexes using the Directory Server Console:

1. On the Directory Server Console, select the Configuration tab.
2. Select Database in the navigation tree, and then select the Indexes tab in the right pane.
3. Remove the index by clearing the checkbox under the type of index you want to remove in the same row as the attribute name.

That is, to remove the presence index for the commonName (cn) attribute, clear the Presence checkbox in the same row as the cn attribute in the left column.

If you want to remove all indexes maintained for a particular attribute, select the attribute's cell under Attribute Name and click Delete Attribute.

For example, to remove all commonName indexing, delete cn from the Attribute Name column by selecting the cell and then clicking Delete Attribute.

4. Click Save.

   The index is no longer maintained for the server, although any index files that were created previous to the deletion are not deleted. To eliminate these old index files, export and reimport your database as described in "Managing Databases Using LDIF" on page 70.

Removing Standard Indexes Using the Command Line

You can remove any indexes you have created, indexes used by other Netscape servers (such as Messaging or Calendar server), and default indexes that are not also system indexes using the command line, or by using ldapdelete. For information on deleting entries using the ldapdelete, see "Deleting Entries Using ldapdelete" on page 247.

To remove indexes using the command line:

1. Stop the server. See "Starting and Stopping the Directory Server" on page 29 for information.

2. Edit slapd.ldbm.conf to remove the index.

   For example, suppose you are maintaining an equality, substring, and presence index for the commonName (cn) attribute. Then slapd.ldbm.conf contains the following line:

   ```
   index cn pres,sub,eq
   ```

If you wanted to remove the substring index, then edit the line so that it reads:

```
index cn pres,eq
```

3. Start your server.

   The index is no longer maintained for the server, although any index files that were created previous to the deletion are not deleted. To eliminate these old index files, export and reimport your database as described in “Managing Databases Using LDIF” on page 70.

Using Browsing Indexes

Browsing indexes can significantly speed up the display of large lists of entries on the Directory Server Console. There are some performance drawbacks to creating any kind of index, so use this feature carefully. See “The Cost of Indexing” on page 180 for information. You might want to create a browsing index on branch points in your directory that contain a large number of entries, for example, the `ou=people` branch. This section describes these topics:

Creating Browsing Indexes

You create browsing indexes through the Directory Server Console on the Directory tab. This process will place your database in read-only mode while the server creates the index. While the database is in read-only mode, you cannot add, modify, or delete entries in the directory.

To create a browsing index:

1. Log in to the Netscape Console using a bind DN with sufficient access permissions to modify the directory, for example, the directory manager.
2. On the Directory Server Console, select the Directory tab.
3. Right-click the entry for which you want to create the index in the navigation tree, for example, `ou=people`, and select Create Browsing Index from the pop-up menu.

The server places the database in read-only mode and creates the index. While the database is in read-only mode, you cannot add, modify, or delete entries in the directory. Once complete, the new index is immediately active for any new data that you might add to your directory. You do not have to restart your server.

Removing Browsing Indexes

You can remove browsing indexes by deleting the index entry either by using `ldapdelete` or through the Directory Server Console. For information on deleting entries using the command-line, see “Deleting Entries Using ldapdelete” on page 247.

To remove a browsing index from the server console:

1. On the Directory Server Console, select the Database tab.
2. Right-click the entry from which you want to delete the index in the navigation tree, for example, `ou=people`, and select Delete Browsing Index from the pop-up menu.
3. The index is no longer maintained for the server, although any index files that were created previous to the deletion are not deleted. To eliminate these old index files, export and reimport your database as described in “Managing Databases Using LDIF” on page 70.

Managing All IDs Threshold

Each index that the directory server uses is composed of a table of index keys and matching entry ID lists. That is, for each index key there is a list of directory entry IDs that match the key. This entry ID list is used by the directory server to build a list of candidate entries that may match a specified search filter (see “The Searching Algorithm” on page 176 for details).

For each entry ID list there is a size limit. This size limit is globally applied to every index key managed by the server, and it is called the All IDs threshold. When the size of an individual ID list reaches this boundary, the server replaces that ID list with an All IDs token.

The All IDs token causes the server to assume that all directory entries match the index key. In effect, the All IDs token causes the server to behave as if no index was available for that type of search. The assumption is that some other aspect of the search filter will allow the server to narrow its candidate list before processing the list.

The following sections examine the benefits and drawbacks of the All IDs mechanism. They also give advice for the tuning of the All IDs threshold.

Benefits of the All IDs Mechanism

The All IDs mechanism is an important mechanism for improving search performance in those cases where the search results would be most if not all directory entries (for example, searches such as `cn=*`). By assuming All IDs, the directory server

- does not have to maintain infinitely increasing entry ID list. This minimizes your directory server’s disk space usage.
- does not have to load unnecessarily large entry ID lists into memory in response to search filters that should result in all directory entries anyway.

  This increases search performance by reducing large disk reads.
- does not require ever increasing amounts of RAM to hold in memory unnecessarily large entry ID lists.

Drawbacks of the All IDs Mechanism

Performance problems can occur if the All IDs threshold is set either too low (this is the most common problem) or too high for your directory's size.

When All IDs Threshold is Too Low

If the All IDs threshold is set too low, then too many index keys will be set to All IDs. This can result in too many directory searches examining every entry in your directory. Since this scenario is most likely to occur when your directory grows to be very large (millions of entries), the performance hit on searches can be excessive.

For example, suppose you are managing an equality index on the common name attribute. One of the index keys stored in your cn index might therefore be `cn=Pete`. The corresponding entry ID list would then contain the ID number of every entry containing a common name attribute that is set to `Pete`.

This is a very useful index to maintain because only a small fraction of the entries in your directory will include `cn=Pete`. Performance for searches that use a `cn=Pete` filter will be vastly improved because only that small fraction of entry IDs needs to be examined when servicing the search request.

However, over time your directory may continue to grow. As it does, more and more Petes may be added, but at the same relatively small proportion of total directory entries. Eventually, the `cn=Pete` entry ID list can become quite large, but it will still be a list that is necessary for search performance. If your directory grows large enough that so many `cn=Pete` entries are added that the All IDs threshold is met, then the `cn=Pete` entry ID list is replaced with an All IDs token. Suddenly, every time you search for `cn=Pete`, the directory server will examine every single entry in the directory in response to the search request.

Of course, for a large enough database size, the All IDs token will be set for a large percentage of all index keys and your search performance will severely degrade.

When All IDs Threshold is Too High

Setting the All IDs threshold to be too high can also cause performance problems. An excessively high All IDs threshold results in large entry ID lists that must be maintained and loaded into memory when servicing search requests. An excessively high All IDs threshold can eliminate all of the benefits of the All IDs mechanism (see “Benefits of the All IDs Mechanism” on page 197 for details).

All IDs Threshold Tuning Advice

Be careful when changing the default All IDs threshold value for your server. If you change the threshold to an inappropriate value, you can hurt your server performance far more than help it.

If your directory is reasonably stable in size, set the All IDs threshold to approximately 5 percent of the total number of entries stored in your directory. That is, if you have 1,000,000 entries in your directory, set the All IDs threshold to 50,000.

If, however, you are planning to add large numbers of entries to your directory in the near future, you probably need to give more thought to your All IDs threshold value. Consider the following:

- Changing All IDs threshold means that you have to rebuild your database. This is a potentially expensive operation, especially for directories that contain millions of entries.
- While setting the All IDs threshold to 5 percent of your directory size is the best value, you should not see serious performance problems if your All IDs threshold is as little as 0.5 percent of your current directory size and as great as 50 percent of your current directory size. That is, you can safely move an order of magnitude in either directory of 5 percent of your directory size. However, you should try to stay as close to the 5 percent rule as possible.

The trick, therefore, is balancing your current directory needs against future expansion plans while trying to avoid changing the All IDs threshold somewhere down the road (which requires a database rebuild).

Suppose, for example, that your current directory is 50,000 entries in size. However, in the next few years you expect to grow your directory to 1,000,000 entries. If you set your All IDs threshold to 5 percent of 50,000 (2500), then when your directory grows to 1,000,000 entries you will have a performance problem because 2500 is far too low for a 1,000,000 entry database (the lower bound for a 1,000,000 entry database is .5 percent of 1,000,000, or 5000).

In this situation, you can do one of the following:

- set the All IDs threshold to the current best value (2500), and plan on rebuilding your database when your directory becomes large enough to warrant it. A database rebuild means shutting down your directory for however long the rebuild takes, or at least putting your directory into read-only mode. It also means reinitializing any consumer servers that your directory server is replicating entries to.
- Find a value that is a bit high for your current needs but that will work well for your future needs. For example, try setting the All IDs threshold to 20,000—that is 40 percent of 50,000 (which puts it within range of your current directory needs) and 2 percent of 1,000,000 (which puts it within range of your future directory needs).

Which strategy you should choose depends on the situation at your site. You should consider the cost of rebuilding your database (and all associated consumer servers) versus potential performance impacts as your All IDs threshold value moves away from the ideal setting of 5 percent. While these impacts should be slight, they may be noticeable for extremely loaded directories. Also consider how quickly your directory will grow and how long you it will take you to increase your directory size. If it will take years to grow, then it may be worth planning on a database rebuild. If, however, it will take only months to increase your directory size by an order of magnitude or greater, then consider ways to set the All IDs threshold such that you minimize the time between database rebuilds.

Default All IDs Threshold Value

By default, the directory server is set to an All IDs threshold of 4000. This value is ideal for a directory size of 80,000 entries. According to the advice given in the previous section, this default value is acceptable for directories between 8000 and 800,000 entries in size. If your anticipated directory size falls outside this range, change your All IDs threshold before populating your database.

Symptoms of an Inappropriate All IDs Threshold Value

When your All IDs threshold is set incorrectly, you will see poor search performance. However, poor search performance can be caused by other factors. For example:

- Your users are performing a lot of searches for which you are not maintaining an index.
- Your database cache size and entry cache size may be set incorrectly (see “Maximum Entries in Cache” on page 482 and “Maximum Cache Size” on page 482 for more information).

Carefully examine these possibilities first before changing your All IDs threshold value.

If you think that your server is suffering from an All IDs threshold that is too low, look in your access log. (See “Access Log” on page 264 for more information.) Any searches that resulted in an All IDs situation will contain the `notes=U` flag. `notes=U` will be returned for

- searches for which you are not maintaining an index
- searches for which an ID list is not maintained because the All IDs threshold value has been reached for that index key

To determine whether the search result belonged to a search that should have been indexed, you have to match the `conn` and `op` values from the RESULT line to a previous SRCH line in your access log file. The SRCH line will show the search filter that was used for the search request. If you have an index for the specified search filter, then the `notes=U` flag is caused by an All IDs flag that has been set for the index key.

```
[24/July/1998:15:12:20 -0800] conn=2 op=1 SRCH base="o=airius.com"
scope=0 filter="(cn=pete)”

[24/July/1998:15:12:20 -0800] conn=2 op=1 RESULT err=0 tag=101
nentries=10000 notes=U
```

Changing the All IDs Threshold Value

To change the All IDs threshold value for your server:

1. Shut down your directory server.
2. Export your directory database to LDIF using the command line. For more information see “Exporting to LDIF From the Command Line” on page 71.
3. Edit `<NSHOME>/slapd-<server ID>/config/slapd.ldbm.conf` with the text editor of your choice.
4. If the `allidsthreshold` parameter is in the file, change it to the desired setting. If `allidsthreshold` is not in the file, add it to the file anywhere after the line beginning with `plugin database`. For more information on the `allidsthreshold` parameter, see “All IDs Threshold” on page 474.
5. Create your database using the command-line. See “Importing LDIF From the Command Line” on page 77 for information.
6. If you are increasing your All IDs threshold value, examine your database cache size.

   Increasing your All IDs threshold can result in larger memory requirements due to larger entry ID lists. The increase in memory requirement will differ depending on the number and types of indexes that you are maintaining, but it will never be larger than the factor by which you increased `allidsthreshold`. That is, if you double `allidsthreshold` then you should not have to increase your database cache size to more than double its current value.

   Increasing your database cache size by the same factor as you increased `allidsthreshold` is an extreme measure. If you have the physical memory available, try increasing your database cache size by a factor that is 25 percent of your `allidsthreshold` increase (that is, if you doubled `allidsthreshold`, then increase your database cache size by 50 percent). Then, if necessary, slowly increase your cache size until you are satisfied with your server’s performance.

You set your database cache size using the `dbcachesize` `slapd.ldbm.conf` parameter. For more information, see “Maximum Cache Size” on page 482.

7. Restart your directory server.

Chapter

8

Finding Directory Entries

You can find entries in your directory using any LDAP client. Most clients provide some form of a search interface that allows you to easily search the directory and retrieve entry information.

Note You cannot search the directory unless the appropriate access control has been set in your directory. For information on setting access control in your directory, see Chapter 5, "Managing Access Control."

The Netscape Directory Server comes with the following LDAP clients that allow you to search your directory:

- Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* for information.
- Directory server gateway. See the online help available through the gateway for information.
- Netscape Directory Express. See the online help for this product for information.
- Directory tab of the Directory Server Console.
- `ldapsearch` command-line utility.

With most LDAP clients you use to search your directory, you can use search filters to help locate entries.

This chapter covers the following topics:

Finding Entries Using the Server Console

You can use the Directory tab of the Directory Server Console to browse through the contents of the directory tree and search for specific entries in the directory.

1. Make sure the Directory Server is running.
2. Bind to the directory by logging in to the Directory Server Console.

   See "Binding to the Directory From Netscape Console" on page 27 for specific instructions.
3. On the Directory Server Console, select the Directory tab.

   Depending on the DN you used to authenticate to the directory, this tab displays the contents of the directory that you have access permissions to view. You can browse through the contents of the tree or right-click an entry and select Search from the pop-up menu. See the online help available through the Search dialog box for information on using this feature.

Warning Netscape strongly recommends that you do not directly modify the contents of the `o=NetscapeRoot` suffix using the Directory tab unless instructed to do so by Netscape Technical Support.

LDAP Search Filters

Search filters select the entries to be returned for a search operation. They are most commonly used with the `ldapsearch` command-line utility. When you use `ldapsearch`, you can place multiple search filters in a file, with each filter on a separate line in the file, or you can specify a search filter directly on the command-line.

For example, the following filter specifies a search for a common name equal to Babs Jensen:

```
cn=babs jensen
```

This search filter returns all entries that contain a common name equal to Babs Jensen. Searches for common name values are case-insensitive.

Any language tagged values associated with the common name attribute are also returned. Thus, the following two attribute values both match this filter:

```
cn: babs jensen
```

```
cn;lang-fr: babs jensen
```

For a list of all the supported language tags, see Table B.1 on page 495.

Search Filter Syntax

The basic syntax of a search filter is `<attribute><operator><value>`. For example:

```
buildingname>=alpha
```

In this example, `buildingname` is the attribute, `>=` is the operator, and `alpha` is the value. You can also define filters that use different attributes combined together using boolean operators. The following sections describe search filters in detail.

Using Attributes in Search Filters

When searching for an entry, you can specify attributes associated with that type of entry. For example, when you search for entries about people, you can use the `cn` attribute to search for people with a specific common name.

Examples of attributes for entries about people might include:

- `cn` (the person's common name)
- `sn` (the person's surname, or last name, or family name)
- `telephonenumber` (the person's telephone number)
- `buildingname` (the name of the building in which the person resides)
- `l` (the location where you can find the person)

For a listing of the attributes associated with types of entries, see the *Netscape Directory Server Schema Reference Guide.*

Using Operators in Search Filters

A search filter operator can be one of those listed in Table 8.1:

Table 8.1 Search filter operators

Search type	Operator	Description
Equality	=	Returns entries containing attribute values that exactly match the specified value. For example, `cn=Bob Johnson`
Substring	=<string>*<string>	Returns entries containing attributes containing the specified substring. For example, `cn=Bob*` `cn=*Johnson` `cn=*John*` `cn=B*John` (The "*" indicates zero (0) or more characters.)

Table 8.1 Search filter operators (Continued)

Search type	Operator	Description
Greater than or equal to	>=	Returns entries containing attributes that are greater than or equal to the specified value. For example, `buildingname >= alpha`
Less than or equal to	<=	Returns entries containing attributes that are less than or equal to the specified value. For example, `buildingname <= alpha`

Presence	=*	Returns entries containing one or more values for the specified attribute. For example,
		`cn=*`
		`telephonenumber=*`
		`manager=*`

Approximate	~=	Returns entries containing the specified attribute with a value that is approximately equal to the value specified in the search filter. For example,
		`cn~=suret`
		`l~=san fransico`
		could return
		`cn=sarette`
		`l=san francisco`

Note In addition to these search filters, you can specify special filters to work with a preferred language collation order. For information on how to search a directory with international character sets, see “Searching an Internationalized Directory” on page 222.

Using Compound Search Filters

Multiple search filter components can be combined using Boolean operators expressed in prefix notation as follows:

```
(<boolean-operator>(<filter>)(<filter>)(<filter>)...))
```

where `<boolean-operator>` is any one of the Boolean operators (described later). In addition, multiple Boolean operators can be nested together to form complex expressions, such as:

```
(<boolean-operator>(<filter>)((<boolean-operator>(<filter>)(<filter>)))
```

Boolean Operators

The Boolean operators available for use with search filters include the following:

Table 8.2 Search filter boolean operators

Operator	Symbol	Description
AND	&	All specified filters must be true for the statement to be true. For example, `(&(filter)(filter)(filter)...)`
OR	|	At least one specified filter must be true for the statement to be true. For example, `(|(filter)(filter)(filter)...)`
NOT	!	The specified statement must not be true for the statement to be true. Only one filter is affected by the `NOT` operator. For example, `(!(filter))`

Boolean expressions are evaluated in the following order:

- Innermost to outermost parenthetical expressions first
- All expressions from left to right

Search Filter Examples

The following filter searches for entries containing one or more values for the manager attribute. This is also known as a presence search:

```
manager=*
```

The following filter searches for entries containing the common name Ray Kultgen. This is also known as an equality search:

```
cn=Ray Kultgen
```

The following filter returns all entries that do not contain the common name Ray Kultgen:

```
(!(cn=Ray Kultgen))
```

The following filter returns all entries that contain a description attribute that contains a substring of X.500:

```
description=*X.500*
```

The following filter returns all entries whose organizational unit is Marketing and whose description field does not contain the substring X.500:

```
(&(ou=Marketing)(!(description=*X.500*)))
```

The following filter returns all entries whose organizational unit is Marketing and that have Julie Fulmer or Cindy Zwaska as a manager:

```
(&(ou=Marketing)(|(manager=cn=Julie
Fulmer,ou=Marketing,o=airius.com)(manager=cn=Cindy
Zwaska,ou=Marketing,o=arius.com)))
```

The following filter returns all entries that do not represent a person:

```
(!(objectClass=person))
```

The following filter returns all entries that do not represent a person and whose common name is similar to printer3b:

```
(&(!(objectClass=person))(cn~=printer3b))
```

Using ldapsearch

You use the `ldapsearch` command-line utility to locate and retrieve directory entries. This utility opens a connection to the specified server using the specified distinguished name and password, and locates entries based on a specified search filter. Search scopes can include a single entry, an entry's immediate subentries, or an entire tree or subtree.

Search results are returned in LDIF format. See Chapter 2, "LDAP Data Interchange Format," for information on LDIF.

For information on where you can find the command line utilities in your directory server installation, see "Finding the Command-Line Utilities" on page 33.

Using Special Characters

When using the `ldapsearch` command-line utility, you may need to specify values that contain characters that have special meaning to the command-line interpreter (such as space [ ], asterisk [*], backslash [\], and so forth). When this situation occurs, enclose the value in quotation marks (""). For example:

```
-D "cn=Barbara Jensen, ou=Product Development, o=airius.com"
```

Depending on which command-line interpreter you use, you should use either single or double quotation marks for this purpose. Refer to your operating system documentation for more information.

In addition, if you are using DNs that contain commas in values, you must escape the commas with a backslash (\). For example:

```
-D "cn=Patricia Fuentes, ou=people, o=Airius Bolivia\, S.A."
```

ldapsearch Command Line Format

When you use `ldapsearch`, you must enter the command using the following format:

```
ldapsearch [<optional parameters>] [<optional search filter>]
[<optional list of attributes>]
```

where

- `<optional parameters>` are a series of command line parameters. These must be specified before the search filter, if any.
- `<optional search filter>` is an LDAP search filter as described in “LDAP Search Filters” on page 207. Do not specify a search filter if you are supplying search filters in a file using the `-f` parameter.
- `<optional list of attributes>` are space separated attributes that reduce the scope of the attributes returned in the search results. This list of attributes must appear after the search filter. For a usage example, see “Displaying Subsets of Attributes” on page 220. If you do not specify a list of attributes, the search returns values for all attributes permitted by the access control set in the directory with the exception of operational attributes.

  If you want operational attributes returned as a result of a search operation, you must explicitly specify them in the search command. To retrieve regular attributes in addition to explicitly specified operational attributes, specify “*” in addition to the operational attributes.

Commonly Used ldapsearch Parameters

The following lists the most commonly used ldapsearch command-line parameters. If you specify a value that contains a space [ ], the value should be surrounded by double quotation marks, for example, `-b "ou=groups, o=airius.com"`.

-b Specifies the starting point for the search. The value specified here must be a distinguished name that currently exists in the database. This parameter is optional if the `LDAP_BASEDN` environment variable has been set to a base DN.

The value specified in this parameter should be provided in double quotation marks. For example: `-b "cn=Barbara Jensen, ou=Product Development, o=airius.com"`.

If you want to search the root DSE entry, specify an empty string here. For example:
`-b ""`

-D Specifies the distinguished name with which to authenticate to the server. This parameter is optional if anonymous access is supported by your server. If specified, this value must be a DN recognized by the directory server, and it must also have the authority to search for the entries. For example, `-D "uid=bjensen, o=airius.com"`.

-h Specifies the hostname or IP address of the machine on which the directory server is installed. If you do not specify a host, `ldapsearch` uses the localhost. For example, `-h mozilla`.

-l Specifies the maximum number of seconds to wait for a search request to complete. Regardless of the value specified here, `ldapsearch` will never wait longer than is allowed by the server's "Time Limit" parameter. For example, `-l 300`. The default value for the Time Limit parameter is 3,600 seconds.

-p Specifies the TCP port number that the directory server uses. For example, `-p 1049`. The default is 389. If `-Z` is used, the default is 636.

-s Specifies the scope of the search. The scope can be one of the following:

- `base`—Search only the entry specified in the `-b` option or defined by the `LDAP_BASEDN` environment variable.
- `one`—Search only the immediate children of the entry specified in the `-b` parameter. Only the children are searched; the actual entry specified in the `-b` parameter is not searched.
- `sub`—Search the entry specified in the `-b` parameter and all of its descendants. That is, perform a subtree search starting at the point identified in the `-b` parameter. This is the default.

-w Specifies the password associated with the distinguished name that is specified in the `-D` option. If you do not specify this parameter, anonymous access is used. For example, `-w diner89&2`.

-x Specifies that the search results are sorted on the server rather than on the client. This is useful if you want to sort according to a matching rule, as with an international search. In general, it is faster to sort on the server rather than on the client.

-z Specifies the maximum number of entries to return in response to a search request. For example, `-z 1000`. Normally, regardless of the value specified here, `ldapsearch` never returns more entries than the number allowed by the server's "Size Limit" parameter. However, you can override this limitation by

binding as the root DN when using this command-line argument. This is because, when you bind as the root DN, this parameter defaults to zero (0). The default value for the Size Limit parameter is 2,000 entries.

SSL Parameters

You can use the following command-line parameters to specify that `ldapsearch` use LDAPS when communicating with your SSL-enabled directory server. You also use these parameters if you want to use certificate-based authentication. These parameters are valid only when LDAPS has been turned on and configured for your Directory Server. For more information on certificate-based authentication, see “Using Certificate-Based Authentication” on page 311. For information on creating a certificate database for use with LDAP clients, see “Creating Certificate Databases for LDAP Clients” on page 313.

Make sure that you specify your directory server's encrypted port, using the -p argument, when you use these parameters.

-I **FORTEZZA Only.** Specifies the personal identification number (PIN) associated with the FORTEZZA crypto card and certificate you specified in the `-Q` parameter. For example, 1234.

-K Specifies the name of the certificate key used for certificate-based client authentication. For example, `-K Server-Key`.

-m Specifies the path to the security module database. For example, `<`*`NSHOME`*`>/netscape/secmodule.db`. You only need to specify this option if the security module database is in a different directory from the certificate database itself.

-N Specifies the certificate name to use for certificate-based client authentication. For example, `-N "Server-Cert"`. If this option is specified, then the `-Z`, `-K`, `-P`, and `-W` parameters are required. Also, if this option is specified, then the `-D` and `-w` parameters must not be specified, or certificate-based authentication will not occur and the bind operation will use the authentication credentials specified on `-D` and `-w`.

-P Specifies the path and filename of the certificate database of the client. This parameter is used only with the `-Z` parameter. When used on a machine where an SSL-enabled version of Netscape Communicator is configured, the path specified on this option can be that of the certificate database for

Communicator. For example, `-P c:\security\cert.db`. The client security files can also be stored on the directory server in the `<NSHOME>/alias` directory. In this case, the `-P` parameter would call out a path and filename similar to the following: `-P c:\Netscape\Server4\alias\client-cert.db`.

-Q **FORTEZZA Only.** Specifies the number of the slot into which you plugged your FORTEZZA crypto card and, optionally, the name of the FORTEZZA certificate you want to use. The slot number and certificate name are separated by a colon. For example, if you plugged your crypto card into slot 2 and want to use the certificate named doe, you would specify the following: `-Q 2:doe`.

-W Specifies the password for the certificate database identified in the `-P` parameter. For example, `-W serverpassword`.

-X **FORTEZZA Only.** Specifies the path and filename of the compromised key list (CKL).

-Z Specifies that SSL is to be used for the search request.

Additional ldapsearch Parameters

To further customize a search, use the following optional parameters:

-A Specifies that the search retrieve the attributes only, not the attribute values. This parameter is useful if you just want to determine if an attribute is present for an entry and you are not interested in the value.

-a Specifies how alias dereferencing is completed. Value can be “never,” “always,” “search,” or “find.” Default value is “never.”

-B Print binary values. Specifies that binary values stored in the directory should be printed in the search output. If you use `-B` and `-o` together, then the binary data will not use base 64 encoding.

-F Specify a different separator. This option can only be used with `-o`. This parameter allows you to specify a separator other than a colon “:” to separate an attribute name from the corresponding value. For example, `-F +`

-f Specifies the file containing the search filter(s) to be used in the search. For example, `-f search_filters`. Search filters are described in “LDAP Search Filters” on page 207. Omit this parameter if you want to supply a search filter directly to the command line.

-G Virtual list search. Allows you to specify the number of entries before or after the search target, and the index or value of the first entry returned. For example, if you are sorting by surname, `-G 20:30:johnson` returns the first entry with a surname equal to or less than johnson, in addition to 20 entries that come before it and 30 entries that come after it. If there are fewer matching entries in the directory than the "before" or "after" number requested by the search, all available entries before/after the search target that match the search criteria are returned.

-i Character set. Specifies the character set to use for command line input. The default is the character set specified in the `LANG` environment variable. You might want to use this parameter to perform the conversion from the specified character set to UTF8, thus overriding the environment variable setting.

Using this argument, you can input the bind DN, base DN, and the search filter pattern in the specified character set. `ldapsearch` converts the input from these arguments before it processes the search request. For example, `-i no` indicates that the bind DN, base DN, and search filter are provided in Norwegian.

This argument only affects the command-line input, that is, if you specify a file containing a search filter (with the `-f` parameter) `ldapsearch` will not convert the data in the file.

-k Conversion routines directory. If you want to specify a sort language that is not supported by default in this release of the directory server, for example, one obtained from a later release of the LDAP SDK, you need to supply the directory in which you store the conversion routines. You can view the list of supported languages in Table B.2 on page 497.

When performing the search, the server looks in the current working directory. However, if the conversion routines are not in the current working directory you need to specify this option when using `ldapsearch`. The conversion routines directory is located by default in `<NSHOME>/<ServerID>/lib/nls`.

-M Manage smart referrals. Causes the server to not return the smart referral contained on the entry, but to instead return the actual entry containing the referral. Use this parameter if you are attempting to search for entries that contain smart referrals. For more information about smart referrals, see "Creating and Changing Smart Referrals" on page 360.

-n Specifies that the search is not to be actually performed, but that ldapsearch is to show what it would do with the specified input.

-O Specifies the maximum number of referral hops `ldapsearch` should automatically follow. For example, `-O 2`.

-o Specifies that the output for individual values be formatted without line breaks and that equal signs "=" be used to separate attribute names from values. This argument produces output in a non-LDIF format.

-R Specifies that referrals are not to be followed automatically. By default, referrals are followed automatically.

-S Specifies the attribute to use as the sort criteria. For example, `-S sn`. You can use multiple -S arguments if you want to further define the sort order. In the following example, the search results will be sorted first by surname and then by given name:

```
-S sn -S givenname
```

The default is not to sort the returned entries.

-T Specifies that no line breaks should be used within individual values in the search results.

-t Specifies that the results be written to a set of temporary files. When you use this option, each attribute value is placed in a separate file within the system temporary directory. No base64 encoding is performed on the values, regardless of the content.

-u Specifies that the user-friendly form of the distinguished name be used in the output.

-v Specifies that the utility is to run in verbose mode.

-V Specifies the LDAP version number to be used on the search. For example, `-V 2`. LDAP v3 is the default. You cannot perform an LDAP v3 search against a directory server that only supports LDAP v2. Only use LDAP v2 when connecting to LDAP v2 servers, such as Netscape Directory Server 1.x.

-y Specifies the proxy DN to use for the search. This argument is provided for testing purposes. For more information about proxied authorization, see "Overview of Proxied Authorization" on page 158.

ldapsearch Examples

For the following examples, suppose the following are true:

- You want to perform a search of all entries in the directory.
- You have configured your directory to support anonymous access for search and read. In this case, you do not have to specify any bind information in order to perform the search. For more information on anonymous access, see “Anonymous Access” on page 105.
- The server is located on hostname `mozilla`.
- The server uses port number 389. Since this is the default port, you do not have to identify the port number on the search request.
- The suffix under which all data is stored is `o=airius.com`.

Returning All Entries

Given the previous information, the following call will return all entries in the directory:

```
ldapsearch -h mozilla -b "o=airius.com" -s sub "objectclass=*"
```

“objectclass=*” is a search filter that matches any entry in the directory.

Specifying Search Filters on the Command Line

You can specify a search filter directly on the call to the command line. If you do this, be sure to enclose your filter in quotation marks (“filter”). Also, do not specify the `-f` parameter. For example:

```
ldapsearch -h mozilla -b "o=airius.com" "cn=babs jensen"
```

Searching the root DSE Entry

Among other things, the root Directory Server Entry (root DSE) contains a list of all the suffixes supported by the local directory server. You can search this entry by supplying a search base of “”. You must also specify a search scope of `base` and a filter of `"objectclass=*"`. For example:

```
ldapsearch -h mozilla -b "" -s base "objectclass=*"
```

Searching the Schema Entry

The Netscape Directory Server stores all directory server schema in a special directory tree who's suffix is `cn=schema`. This tree contains a single entry (`cn=schema`), and this entry contains information on every object class and attribute defined for your directory server.

You can examine the contents of this entry as follows:

```
ldapsearch -h mozilla -b "cn=schema" -s base "objectclass=*"
```

Using LDAP_BASEDN

To make searching easier, you can set your search base using the `LDAP_BASEDN` environment variable. Doing this allows you to skip specifying the search base with the `-b` parameter (for information on how to set environment variables, see the documentation for your operating system).

Typically, you set `LDAP_BASEDN` to your directory's suffix value. Since your directory suffix is equal to the root, or topmost, entry in your directory, this causes all searches to begin from your directory's root entry.

For example, suppose you have set LDAP_BASEDN to `o=airius.com`. Then to search for `cn=babs jensen` in your directory use the following command-line call:

```
ldapsearch -h mozilla "cn=babs jensen"
```

In this example, the default scope of “sub” is used.

Displaying Subsets of Attributes

ldapsearch returns all search results in LDIF format. By default, ldapsearch returns the entry's distinguished name and all of the attributes that you are allowed to read (you can set up the directory access control such that you are allowed to read only a subset of the attributes on any given directory entry), with the exception of operational attributes. If you want operational attributes returned as a result of a search operation, you must explicitly specify them in the search command.

Suppose you do not want to see all of the attributes returned on the search results. In this case, you can limit the returned attributes to just a few specific attributes by specifying those attributes on the command line immediately after the search filter. For example, to show the `cn` and `sn` attributes for every entry in the directory, use the following command-line call:

```
ldapsearch -h mozilla "objectclass=*" sn cn
```

This example assumes you set your search base with `LDAP_BASEDN`.

Specifying Search Filters Using a File

You can enter search filters into a file instead of entering them on the command line. When you do this, specify each search filter on a separate line in the file. `ldapsearch` will run each search in order until the last search filter is found in the file. That is, if you enter

```
sn=Francis
givenname=Richard
```

into the file, then `ldapsearch` first finds all the entries who's surname is Francis, and then all the entries whose givenname is Richard. If an entry is found that matches both search criteria, then that entry is returned twice.

For example, suppose you specified the previous search filters in a file named `searchdb`, and you set your search base using `LDAP_BASEDN`. Then the following returns all the entries that match either search filter:

```
ldapsearch -h mozilla -f searchdb
```

You can limit the set of attributes returned here by appending the attribute names that you want to see at the end of the search line. For example, the following performs both searches, but only returns the entry's DN and each entry's `givenname` and `sn` attributes:

```
ldapsearch -h mozilla -f searchdb sn givenname
```

Specifying DNs that Contain Commas in Search Filters

When a DN within a search filter contains a comma as part of its value, you must escape the comma with a backslash (\). For example, to find everyone in the Airius Bolivia, S.A. subtree, you would use the following command:

```
ldapsearch -h mozilla -s base -b "o=Airius Bolivia\, S.A."
"objectclass=*"
```

Searching an Internationalized Directory

When you perform search operations, you can request that the directory server sort the results based on any language for which the server has a supporting collation order. For a listing of the collation orders supported by the directory server, see “Identifying Supported Locales” on page 495.

Note When performing internationalized searches, you must perform an LDAP v3 search; do not therefore specify the `-V2` parameter on the call to `ldapsearch`.

This section focuses on the matching rule filter portion of the `ldapsearch` syntax. For more information on general `ldapsearch` syntax, see “LDAP Search Filters” on page 207. For information on searching internationalized directories using the Users and Groups portion of the Netscape Console, refer to the online help or *Managing Servers with Netscape Console*.

Supported Search Types

The directory server supports the following types of international searches:

- equality (=)
- substring (*)
- greater than (>)

- greater than or equal to (>=)
- less than (<)
- less than or equal to (<=)

Approximate, or phonetic, and presence searches are supported only in English.

As with a regular `ldapsearch` search operation, an international search uses operators to define the type of search. However, when invoking an international search, you can either use the standard operators (`=, >=, >, <, <=`) in the value portion of the search string, or you can use a special type of operator, called a suffix (not to be confused with the directory suffix), in the matching rule portion of the filter to define the search type. Table 8.3 summarizes each type of search, the operator, and the equivalent suffix.

Table 8.3 Search types, operators, and suffixes

Search Type	Operator	Suffix
Less than	<	.1
Less than or equal to	<=	.2
Equality	=	.3
Greater than or equal to	>=	.4
Greater than	>	.5
Substring	*	.6

Matching Rule Filter Syntax

A matching rule provides special guidelines for how strings are to be compared in a search operation. In an international search, the matching rule tells the system what collation order and operator to use when performing the search operation. For example, a matching rule in an international search might tell the server to search for attribute values that come at or after llama in the Spanish collation order. The syntax of the matching rule filter is as follows:

```
<attr>:<matchingRule>:=<value>
```

where

- `<attr>` is the attribute of entries you're searching for, such as cn or mail
- `<matchingRule>` is a string that identifies either the collation order or the collation order and a relational operator, depending on the format you prefer. For a discussion of matching rule formats, see "Matching Rule Formats" on page 224.
- `<value>` is either the attribute value you want to search for or a relational operator plus the attribute value you want to search for; the syntax of the value portion of the filter depends on the matching rule format you use.

Matching Rule Formats

There are several ways that you can represent the matching rule portion of a search filter, the one you use is a matter of personal preference. The matching rule can be any of the following:

- The OID for the collation order on which you want to base your search.
- The language tag associated with the collation order on which you want to base your search.
- The OID for the collation order on which you want to base your search, and a suffix that represents a relational operator.
- The language tag associated with the collation order on which you want to base your search, and a suffix that represents a relational operator.

The syntax for each of these options is discussed in the following sections.

Using an OID for the Matching Rule

Each locale supported by the directory server has an associated collation order OID. For a list of locales supported by the directory server and their associated OIDs, see Table B.1 on page 495. You can use the collation order OID in the matching rule portion of the matching rule filter as follows:

```
<attr>:<OID>:=(<relational operator><space><value>)
```

In this case, the relational operator is included in the value portion of the string, separated from the value by a single space. For example, to search for all `departmentNumber` attributes that are at or after N4709 in the Swedish collation order, use the following filter:

```
departmentNumber:2.16.840.1.113730.3.3.2.46.1:=>= N4709
```

Using a Language Tag for the Matching Rule

Each locale supported by the directory server has an associated language tag. For a list of locales supported by the directory server and their associated language tags, see Table B.1 on page 495. You can use the language tag in the matching rule portion of the matching rule filter as follows:

```
<attr>:<language-tag>:=(<relational operator><space><value>)
```

In this case, the relational operator is included in the value portion of the string, separated from the value by a single space. For example, to search the directory for all `description` attributes with a value of estudiante using the Spanish collation order, use the following filter:

```
cn:es:== estudiante
```

Using an OID and Suffix for the Matching Rule

As an alternative to using a relational operator + value pair, you can append a suffix that represents a specific operator to the OID in the matching rule portion of the filter. For a list of locales supported by the directory server and their associated OIDs, see Table B.1 on page 495. For a list of relational operators and their equivalent suffixes, see Table 8.3. You combine the OID and suffix as follows:

```
<attr>:<OID>+<suffix>:=<value>
```

For example, to search for `businessCategory` attributes with the value softwareproduckte in the German collation order, use the following filter:

```
businessCategory:2.16.840.1.113730.3.3.2.7.1.3:=softwarep
rodukte
```

The `.3` in the previous example is the equality suffix.

Using a Language Tag and Suffix for the Matching Rule

As an alternative to using a relational operator + value pair, you can append a suffix that represents a specific operator to the language tag in the matching rule portion of the filter. For a list of locales supported by the directory server and their associated language tags, see Table B.1 on page 495. For a list of relational operators and their equivalent suffixes, see Table 8.3. You combine the language tag and suffix as follows:

```
<attr>:<language-tag>+<suffix>:=<value>
```

For example, to search for all surnames that come at or after La Salle in the French collation order, use the following filter:

```
sn:fr.4:=La Salle
```

Using Wildcards in Matching Rule Filters

When performing a substring search using a matching rule filter, you can use the asterisk (*) character as a wildcard to represent zero or more characters. For example, to search for an attribute value that starts with the letter “l” and ends with the letter “n,” you would enter a “l*n” in the value portion of the search filter. Similarly, to search for all attribute values beginning with the letter “u,” you would enter a value of “u*” in the value portion of the search filter.

To search for a value that contains the asterisk (*) character, you must escape the * with the designated escape sequence, \5c2a. For example, to search for all employees with `businessCategory` attribute values of Airius*Net product line, enter the following value in the search filter:

```
Airius\2a*Net product line
```

International Search Examples

The following sections show examples of how to perform international searches on directory server data. Each example shows all the possible matching rule filter formats so that you can become familiar with the formats and select the one that works best for you.

Less Than Example

When you perform locale-specific search using the less than operator (<) or suffix (.1), you are searching for all attribute values that come before the given attribute in a specific collation order. For example, to search for all surnames that come before the surname Marquez in the Spanish collation order, you could use any of the following matching rule filters:

```
sn:2.16.840.1.113730.3.3.2.15.1:=< Marquez
sn:es:=< Marquez
sn:2.16.840.1.113730.3.3.2.15.1.1:=Marquez
sn:es.1:=Marquez
```

Less Than or Equal to Example

When you perform locale-specific search using the less than or equal to operator (<=) or suffix (.2), you are searching for all attribute values that come at or before the given attribute in a specific collation order. For example, to search for all room numbers that come at or before room number CZ422 in the Hungarian collation order, you could use any of the following matching rule filters:

```
roomNumber:2.16.840.1.113730.3.3.2.23.1:=<= CZ422
roomNumber:hu:=<= CZ422
roomNumber:2.16.840.1.113730.3.3.2.23.1.2:=CZ422
roomNumber:hu.2:=CZ422
```

Equality Example

When you perform locale-specific search using the equal to operator (=) or suffix (.3), you are searching for all attribute values that match the given attribute in a specific collation order. For example, to search for all `businessCategory` attributes with the value softwareprodukte in the German collation order, you could use any of the following matching rule filters:

```
businessCategory:2.16.840.1.113730.3.3.2.7.1:== softwareprodukte
businessCategory:de:== softwareprodukte
businessCategory:2.16.840.1.113730.3.3.2.7.1.3:=softwareprodukte
businessCategory:de.3:=softwareprodukte
```

Greater Than or Equal to Example

When you perform locale-specific search using the greater than or equal to operator (>=) or suffix (.4), you are searching for all attribute values that come at or after the given attribute in a specific collation order. For example, to search for all localities that come at or after Québec in the French collation order, you could use any of the following matching rule filters:

```
locality:2.16.840.1.113730.3.3.2.18.1:=>= Québec
locality:fr:=>= Québec
locality:2.16.840.1.113730.3.3.2.18.1.4:=Québec
locality:fr.4:=Québec
```

Greater Than Example

When you perform locale-specific search using the greater than operator (>) or suffix (.5), you are searching for all attribute values that come at or before the given attribute in a specific collation order. For example, to search for all mail hosts that come after host schranka4 in the Czechoslovakian collation order, you could use any of the following matching rule filters:

```
mailHost:2.16.840.1.113730.3.3.2.5.1:=> schranka4
mailHost:cs:=> schranka4
mailHost:2.16.840.1.113730.3.3.2.5.1.5:=schranka4
mailHost:cs.5:=schranka4
```

Substring Example

When you perform an international substring search, you are searching for all values that match the given pattern in the specified collation order. For example, to search for all user IDs that end in ming in the Chinese collation order, you could use any of the following matching rule filters:

```
uid:2.16.840.1.113730.3.3.2.49.1:=* *ming
uid:zh:=* *ming
uid:2.16.840.1.113730.3.3.2.49.1.6:=* *ming
uid:zh.6:=* *ming
```

Chapter

9

Managing Directory Entries

In Netscape Directory Server, you add, modify, and delete entries using an LDAP client.

Netscape Directory Server comes with the following LDAP clients that allow you to search and modify your directory:

- Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* for information.
- Directory server gateway. See the online help available through the gateway for information.
- Netscape Directory Express. See the online help for this product for information.
- The Directory tab on the Directory Server Console.
- The following command-line utilities:
  - ldapmodify
  - ldapdelete

This chapter discusses how to use the Directory Server Console and the `ldapmodify` and `ldapdelete` command-line utilities to modify the contents of your directory in the following sections:

Note You cannot modify your directory unless the appropriate access control has been set. For information on setting access control in your directory, see Chapter 5, "Managing Access Control."

Managing Entries Using the Server Console

Use the Directory tab and the Property Editor on the Directory Server Console to add, modify, or delete entries one at a time. You can also use the Directory tab to access the Users and Groups dialog boxes that allow you to manage users, groups, and organizational units. The Users and Groups dialog boxes are also accessible from the Users and Groups area of the Netscape Console. See *Managing Servers with Netscape Console* for more information.

Use the command line to add multiple entries simultaneously. See "Managing Entries Using the Command-Line Utilities" on page 240 for more information.

This section provides information about:

Managing Users, Groups, and Org. Units Using the Server Console

Use the Users and Groups dialog boxes and the Property Editor on the Directory Server Console to add and modify users, groups, and organizational units one at a time. Use LDIF to add or modify multiple entries simultaneously. You can also add or modify users, groups, and organizational units through the Netscape Console Users and Groups tab, the directory server gateway, and through Netscape Directory Express. For information on how to do this, see *Managing Servers with Netscape Console* or the online help for the gateway and Directory Express.

This section provides information about:

- "Adding Users, Groups, and Org. Units Using the Server Console" on page 231.
- "Modifying Users, Groups, and Org. Units Using the Server Console" on page 232.

For information about adding other types of entries using the server console, see "Adding Other Types of Entries Using the Property Editor" on page 234.

Adding Users, Groups, and Org. Units Using the Server Console

To add users, groups, and organizational units one at a time:

1. On the Directory Server Console, select the Directory tab.
2. Right-click the entry in the left pane under which you want to add the user, group, or organizational unit and select New|User, New|Group, or New|Organizational Unit from the pop-up menu.

   The new entry will be created as a child entry of the entry you select. The Create New User, Create New Group, or Create New Organizational Unit dialog box appears.

   Provide the information for the new entry in the dialog box. Click Help for more detailed information about the options on these dialog boxes.

If you want to add attributes or object classes that are not displayed through the Users and Groups dialog boxes, click Advanced to access the Property Editor. For more information on using the Property Editor, see “Using the Property Editor to Manage Entries” on page 232.

3. When you are finished defining the information for the entry, click OK.

Modifying Users, Groups, and Org. Units Using the Server Console

To modify users, groups, and organizational units one at a time:

1. On the Directory Server Console, select the Directory tab.

   The directory contents appear in the left pane.

2. Right-click the entry you want to modify and select Open from the pop-up menu.

   If you selected a user entry, the Edit User dialog box appears. If you selected a group entry, the Edit Group dialog box appears. If you selected an organizational unit, the Edit Entry dialog box appears. Click Help for more detailed information about the options on these dialog boxes.

   If you want to modify attributes or object classes that are not displayed through the Users and Groups dialog boxes, click Advanced to access the Property Editor. For more information on using the Property Editor, see “Using the Property Editor to Manage Entries” on page 232.

   If you selected an entry that is not a user, group, or organizational unit, the property editor appears. See “Using the Property Editor to Manage Entries” on page 232 for more information.

Using the Property Editor to Manage Entries

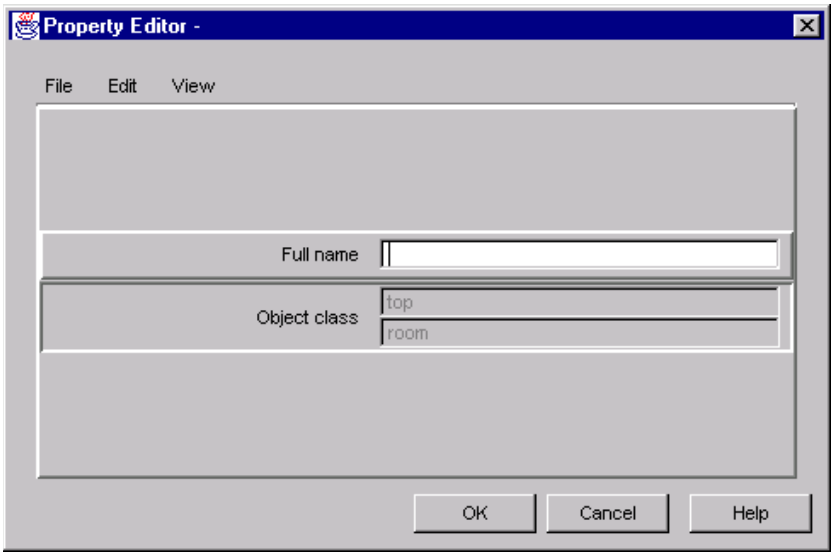
You can use the Property Editor to add or modify attributes and object classes for users, groups, and organizational units. To access the Property Editor from the Users and Groups dialog boxes, click Advanced. See “Managing Users,

Groups, and Org. Units Using the Server Console" on page 231 and *Managing Servers with Netscape Console* for more information about the Users and Groups dialog boxes.

You can also use the Directory tab and the Property Editor to add and modify entries other than users, groups, and organizational units. For example, you can use this method to define entries based on object classes you have created.

The Property Editor contains a list of attributes defined for the object class(es) you selected.

Figure 9.1 Directory Server Console - Property Editor



The following procedures describe how to add and modify entries using the Property Editor.

- "Adding Other Types of Entries Using the Property Editor" on page 234.
- "Adding an Object Class to an Entry Using the Property Editor" on page 235.
- "Removing an Object Class From an Entry Using the Property Editor" on page 235.
- "Adding an Attribute Value to an Entry Using the Property Editor" on page 236.

- “Adding Values to an Attribute Using the Property Editor” on page 236.
- “Removing an Attribute Value From an Entry Using the Property Editor” on page 237.
- “Adding an Attribute Subtype Using the Property Editor” on page 237.

Adding Other Types of Entries Using the Property Editor

To add entries other than users, groups, and organizational units using the Property Editor:

1. On the Directory Server Console, select the Directory tab.
2. Right-click the entry in the left pane to which you want to add the entry and select New | Other from the pop-up menu.

   The New Object dialog box appears.
3. Select the object class(es) with which you want to define the entry.

   To select multiple object classes, use Ctrl+click or Shift+click. Click OK. The Property Editor appears.

   If you select an object class related to a user or group, such as inetOrgPerson or organizationalPerson, the Create New User or Create New Group dialog box appears. See “Adding Users, Groups, and Org. Units Using the Server Console” on page 231 for information about adding users and groups. If you want to further define the new user, group, or organizational unit, click Advanced. (For more information about object classes, see the *Netscape Directory Server Schema Reference Guide*.)
4. Fill in the values for the required attributes and click OK to save the new entry.

Adding an Object Class to an Entry Using the Property Editor

To add an object class to an entry from the Property Editor:

1. On the Directory tab of the Directory Server Console, right-click the entry you want to modify and select Open from the pop-up menu.

   The Property Editor appears.

2. Scroll down the table until you see the `object class` attribute in the left column.

3. Right click `object class` and select Add Value from the pop-up menu.

   The Add Object Class dialog box appears.

4. Select the object class you want to add and click OK to return to the Property Editor.

5. Click OK again when you are finished editing the entry.

Removing an Object Class From an Entry Using the Property Editor

To remove an object class from an entry from the Property Editor:

1. On the Directory tab of the Directory Server Console, right-click the entry you want to modify and select Open from the pop-up menu.

   The Property Editor appears.

2. Scroll down the table until you see the `object class` attribute in the left column.

3. Click the cursor in the text box in the right column that contains the object class you want to remove and select Delete Value from the Edit menu.

4. Click OK when you are finished editing the entry.

Adding an Attribute Value to an Entry Using the Property Editor

Before you can add an attribute value to an entry, the entry must contain an object class that either requires or allows the attribute. See “Adding an Object Class to an Entry Using the Property Editor” on page 235 and *Netscape Directory Server Schema Reference Guide* for more information.

To add an attribute value to an entry from the Property Editor:

1. On the Directory tab of the Directory Server Console, right-click the entry you want to modify and select Open from the pop-up menu.

   The Property Editor appears.

2. Select Add Attribute from the Edit menu.

   The Add Attribute dialog box displays.

3. Select the attribute you want to add in the list and click OK to return to the Property Editor.

4. Scroll through the attributes list until you find the attribute you added.

5. Type the value for the new attribute in the empty text box to the right of the attribute name.

6. Click OK when you are finished editing the entry.

Adding Values to an Attribute Using the Property Editor

You can add more than one value to multi-valued attributes contained within an entry.

To add an additional value using the Property Editor:

1. On the Directory tab of the Directory Server Console, right-click the entry you want to modify and select Open from the pop-up menu.

   The Property Editor appears.

2. Scroll down the table until you see the attribute in the left column.

3. Right click the attribute and select Add Value from the pop-up menu.
4. In the empty text box that appears, type in the name of the new attribute value.
5. Click OK.

Removing an Attribute Value From an Entry Using the Property Editor

To remove an attribute value from an entry from the Property Editor:

1. On the Directory tab of the Directory Server Console, right-click the entry you want to modify and select Open from the pop-up menu.

   The Property Editor appears.
2. Scroll through the attributes list until you find the attribute for which you want to remove a value.
3. Click the cursor in the text box in the right column that contains the attribute value you want to remove and select Delete Value from the Edit menu.

   If you want to remove the entire attribute and all its values from the entry, select Delete Attribute from the Edit menu.
4. Click OK when you are finished editing the entry.

Adding an Attribute Subtype Using the Property Editor

You can add three different kinds of subtypes to attributes contained within an entry; language, binary, and pronunciation.

To add a subtype using the Property Editor:

1. On the Directory tab of the Directory Server Console, right-click the entry you want to modify and select Open from the pop-up menu.

   The Property Editor appears.

2. Select Add Attribute from the Edit menu.

   The Add Attribute dialog box displays.

3. Select the attribute you want to add in the list.

4. To assign a language subtype to the attribute, select it from the Language drop-down list.

   Sometimes a user's name can be more accurately represented in characters of a language other than the default language. For example, Noriko's name is Japanese, and she has indicated on her hiring forms that she prefers that her name be represented by Japanese characters when possible. You can select Japanese as a language subtype for the `givenname` attribute so that other users can search for her Japanese name.

   If you specify a language subtype for an attribute, the subtype is added to the attribute name as follows:

   `<attribute>;lang-<subtype>`

   Where `<attribute>` is the attribute you are adding to the entry and `<subtype>` is the two character abbreviation for the language. See Table B.2 on page 497 for a list of supported language subtypes. For example:

   `givenname;lang-ja`

   You can assign only one language subtype per instance of an attribute in an entry. To assign multiple language subtypes, add another instance of the attribute to the entry and then assign the new language subtype to the copy. For example, `cn;lang-ja;lang-en-GB:Smith` is illegal. Instead, use:

   `cn: lang-ja: <ja_value>`
   cn: lang-en-GB: <en-GB_value>

5. You can also assign one of two other subtypes to any instance of an attribute:

   - Binary—Indicates that the attribute value is binary. For example, `usercertificate;binary`. Although you can store binary data within an attribute that does not contain the `binary` subtype, for example, `jpegphoto`, the `binary` subtype indicates to clients that multiple variants of the attribute type may exist.

   - Pronunciation—Indicates that the attribute value is a phonetic representation. The subtype is added to the attribute name as follows: `<attribute>;phonetic`. This subtype is commonly used in combination with a language subtype for languages that have more than one alphabet, where one is a phonetic representation. You might want to use this with attributes that are expected to contain user names, such as `cn` or `givenname`. For example, `givenname;lang-ja;phonetic` indicates that the attribute value is the phonetic version of the entry's Japanese name.

6. Click OK to return to the Property Editor.

7. When you are finished defining the information for the entry, click OK.

Deleting Entries Using the Server Console

To delete entries using the Server Console:

1. On the Directory Server Console, select the Directory tab.

   The directory contents appear in the left pane.

2. Right-click the entry you want to delete and select Delete from the pop-up menu.

   To select multiple entries, use Ctrl+click or Shift+click and then select Delete from the Edit menu.

   The server deletes the entry or entries immediately. There is no undo.

Managing Entries Using the Command-Line Utilities

The command-line client utilities allow you to manipulate the contents of your directory. The command-line utilities are especially useful for writing scripts to perform bulk management of your directory, or for testing your Directory Server to ensure that it is working correctly (especially if you have changed your access control information) or written a custom client.

This section provides information about:

Using Special Characters

When using the Directory Server command-line client tools, you may need to specify values that contain characters that have special meaning to the command-line interpreter (such as space [ ], asterisk [*], backslash [\], and so forth). When this situation occurs, enclose the value in quotation marks (""). For example:

```
-D "cn=Barbara Jensen, ou=Product Development, o=airius.com"
```

Depending on the command-line utility you use, you should use either single or double quotation marks for this purpose. Refer to your operating system documentation for more information.

In addition, if you are using DNs that contain commas, you must escape the commas with a backslash (\). For example:

```
-D "cn=Patricia Fuentes, ou=people, o=Airius Bolivia\, S.A."
```

Providing Input From the Command Line

`ldapmodify` and `ldapdelete` allow you to provide input both from an input file (using the `-f` parameter), as well as from the command line. If you want to provide input from the command line, do not specify the `-f` parameter when you use these commands.

The tool collects statements you enter in exactly the same way as if they were read from a file. When you finish providing input, enter the character that your shell recognizes as the end of file (EOF) marker. This causes the utility to begin operations based on the input you supplied.

Typically, the EOF escape sequence is one of the following, depending upon the type of machine you use:

- Unix—Almost always control-D (^D)
- Windows NT—Usually control-z followed by <return> (^z<return>)

For example, suppose you wanted to specify some LDIF update statements to `ldapmodify`. Then, on Unix, you would do the following:

```
prompt> ldapmodify -D <bindDN> -w <password> -h <hostname>
> dn: cn=Barry Nixon, ou=people, o=airius.com
> changetype: modify
> delete: telephonenumber
> -
> add: manager
> manager: cn=Harry Cruise, ou=people, o=airius.com
> ^D
prompt>
```

When you add an entry from the command-line or from LDIF, make sure that an entry representing a branch point is created before new entries are created under that branch. For example, if you want to place an entry in a Person and a Group subtree, then create the branch point for those subtrees before creating entries within the subtrees.

For example:

```
dn: o=airius.com
dn: ou=People, o=airius.com
...
<People subtree entries.>
...
dn: ou=Group, o=airius.com
...
<Group subtree entries.>
...
```

Adding Entries Using LDIF

You can use an LDIF file to add multiple entries or to import an entire database. For details, refer to “Importing LDIF From the Server Console” on page 75. You can also add or edit entries using the `ldapmodify` command along with the appropriate LDIF update statements. For details, refer to “Adding and Modifying Entries Using ldapmodify” on page 243. For details on LDIF, see Chapter 2, “LDAP Data Interchange Format.”

To add entries using an LDIF file and the Directory Server Console:

1. Define the entries in an LDIF file.

   LDIF is described in Chapter 2, “LDAP Data Interchange Format.”

2. Import the LDIF file from the Directory Server Console.

   See “Importing Databases From LDIF” on page 74 for information. When you import the LDIF file, select “Append to database” on the Import dialog box so that the server will only import entries that do not currently exist in the directory.

Adding and Modifying Entries Using ldapmodify

You use the `ldapmodify` command-line utility to modify entries in an existing Directory Server database. `ldapmodify` opens a connection to the specified server using the distinguished name and password you supply, and modifies the entries based on LDIF update statements contained in a specified file. Because `ldapmodify` uses LDIF update statements, `ldapmodify` can do everything that `ldapdelete` can do.

For information on where you can find the command-line utilities in your Directory Server installation, see “Finding the Command-Line Utilities” on page 33.

If schema checking is turned on when you use this utility, then the server performs schema checking for the entire entry when it is modified. If the server detects an attribute or object class in the entry that is not known to the server, then the entire modify operation fails. Also, if a required attribute is not present, the modify operation fails. This happens even if the offending object class or attribute is not being modified.

This situation occurs if you ran the Directory Server with schema checking turned off, added unknown object classes or attributes, and then turned schema checking on.

Note You cannot create a root entry (such as `o=airius.com`) using `ldapmodify` unless you bind to the directory as the root DN, for example, `cn=Directory Manager`.

For more information, see “Turning Schema Checking On and Off” on page 58.

This section provides information about:

Commonly Used ldapmodify Parameters

To modify an entry or entries in an existing directory, use the `ldapmodify` command-line utility with the following parameters:

-a Allows you to add LDIF entries to the directory without requiring the `changetype:add` LDIF update statement. This provides a simplified method of adding entries to the directory. This option also allows you to directly add a file created by `ldapsearch`.

-D Specifies the distinguished name with which to authenticate to the server. The value must be a DN recognized by the directory server, and it must also have the authority to modify the entries. For example, `-D uid=bjensen, o=airius.com`. You cannot use this parameter with the -N parameter.

-f Optional parameter that specifies the file containing the LDIF update statements used to define the directory modifications. For example, `-f modify_statements`. If you do not supply this parameter, the update statements are read from `stdin`. For information on supplying LDIF update statements from the command line, refer to “Providing Input From the Command Line” on page 241.

-h Specifies the name of the host on which the server is running. For example, `-h cyclops`.

-p Specifies the port number that the server uses. For example, `-p 1049`. The default is 389. If -Z is used, the default is 636.

-w Specifies the password associated with the distinguished name specified in the -D parameter. For example, `-w mypassword`.

SSL Parameters

You can use the following command-line parameters to specify that `ldapmodify` is to use LDAP over SSL (LDAPS) when communicating with your Directory Server. LDAPS encrypts data during transit. You also use these parameters if you want to use certificate-based authentication. These parameters are valid only when SSL has been turned on and configured for your Directory Server. For more information on certificate-based authentication, see “Using Certificate-Based Authentication” on page 311. For information on creating a certificate database for use with LDAP clients, see “Creating Certificate Databases for LDAP Clients” on page 313.

Make sure that you specify your Directory Server's encrypted port when you use these parameters:

-I **FORTEZZA only.** Specifies the personal identification number (PIN) associated with the FORTEZZA crypto card and certificate you specified in the -Q parameter.

-K Specifies the name of the certificate key used for certificate-based client authentication. For example, `-K Server-Key`.

-N Specifies the certificate name to use for certificate-based client authentication. For example, `-N Server-Cert`. If this parameter is specified, then the -Z, -K, and -W parameters are required. Also, if this parameter is specified, then the -D and -w parameters must not be specified, or certificate-based authentication will not occur and the bind operation will use the authentication credentials specified on -D and -w.

-P Specifies the path and filename of the security files for the client. This parameter is used only with the -Z parameter. When used on a machine where a SSL-enabled web browser is configured, the path specified on this parameter can be pointed to the security database for the web browser. For example, `-P c:\security\cert.db`. You can also store the client security files on the Directory Server in the *<NSHOME>*`/alias` directory. In this case, the -P parameter calls out a path and filename similar to the following:
`-P c:\Netscape\Suitespot\alias\client-cert.db`

-Q **FORTEZZA only.** Specifies the number of the slot into which you plugged your FORTEZZA crypto card and, optionally, the name of the FORTEZZA certificate you want to use. The slot number and certificate name are separated by a colon. For example, if you plugged your crypto card into slot 2 and want to use the certificate named doe, you would specify the following: `-Q 2:doe`.

-W Specifies the password for the certificate database identified on the -P parameter. For example, `-W serverpassword`.

-X **FORTEZZA only.** Specifies the path and filename of the compromised key list (CKL).

-Z Specifies that SSL is to be used for the directory request.

Additional ldapmodify Parameters

The following parameters offer additional functionality:

-b Causes the utility to check every attribute value to determine whether the value is a valid file reference. If the value is a valid file reference, then the content of the referenced file is used as the attribute value. This is often used for specifying a path to a file containing binary data, such as JPEG. For example, if you wanted to add a `jpegPhoto` attribute, then specify the -b parameter on the `ldapmodify` call. In the LDIF you provide to `ldapmodify`, include something like the following:

```
jpegPhoto: /tmp/photo.jpeg
```

`ldapmodify` reads the contents of the `photo.jpeg` file into the `jpegPhoto` attribute that you are placing on the entry.

On Windows NT, the format of this parameter is exactly the same (including the forward slashes, except that you can specify a drive letter. For example:

```
jpegPhoto: c:/tmp/photo.jpeg
```

-c Specifies that the utility run in continuous operation mode. Errors are reported, but the utility continues with modifications. The default is to quit after reporting an error.

-H Lists all available `ldapmodify` parameters.

-M Manage smart referrals. Causes the server to not return the smart referral contained on the entry, but to instead apply the modification request directly to the entry. Use this parameter if you are attempting to add, change, or delete a directory entry that contains a smart referral. For more information about smart referrals, see “Creating and Changing Smart Referrals” on page 360.

-n Specifies that the entries are not to be actually modified, but that `ldapmodify` is to show what it would do with the specified input.

-O Specifies the maximum number of referral hops to follow. For example, `-O 2`.

-R Specifies that referrals are not to be followed automatically.

-v Specifies that the utility is to run in verbose mode.

-V Specifies the LDAP version number to be used on the operation. For example, `-V 2`. LDAP v3 is the default. Note that you can not perform an LDAP v3 operation against a Directory Server that only supports LDAP v2.

-y Specifies the proxy DN to use for the modify operation. This argument is provided for testing purposes. For more information about proxied authorization, see “Overview of Proxied Authorization” on page 158.

ldapmodify Example

Suppose:

- You want to modify entries as specified in the file `modify_statements`.
- You have configured a special entry for the database administrator that has the authority to modify the entries, and that entry has the distinguished name of `cn=Directory Manager, o=airius.com`.
- The database administrator's password is `King-Pin`.
- The server is located on `cyclops`.
- The server uses port number 845.

Then to modify the entries, first specify the appropriate LDIF update statements in the `modify_statements` file, and then enter the following command:

```
ldapmodify -D “cn=Directory Manager, o=airius.com” -w King-Pin -h
cyclops -p 845 -f modify_statements
```

Deleting Entries Using ldapdelete

You use the `ldapdelete` command-line utility to delete entries from an existing Directory Server database. This utility opens a connection to the specified server using the distinguished name and password you provide, and deletes the entry or entries. (For information on where you can find the command-line utilities in your Directory Server installation, see “Finding the Command-Line Utilities” on page 33.)

You can only delete entries at the end of a branch. You cannot delete entries that are branch points in the directory tree. For example, of the following three entries:

```
ou=People, o=airius.com
cn=Paula Simon, ou=People, o=airius.com
cn=Jerry O’Connor, ou=People, o=airius.com
```

you can delete only the last two entries. The entry that identifies the People subtree can be deleted only if no other entries exist below it. If you want to delete `ou=People, o=airius.com`, you must first delete `cn=Paula Simon, ou=People, o=airius.com` and `cn=Jerry O'Connor, ou=People, o=airius.com`.

This section provides information about:

Commonly Used ldapdelete Parameters

To delete an entry or entries from an existing database, use the `ldapdelete` command-line utility with the following parameters:

-D Specifies the distinguished name with which to authenticate to the server. The value must be a DN recognized by the Directory Server, and it must also have the authority to delete the entries. For example, `-D uid=bjensen, o=airius.com`. Access control is described in Chapter 5, "Managing Access Control." If you use the -D parameter, you cannot use the -N parameter.

-h Specifies the name of the host on which the server is running. For example, `-h cyclops`. The default is `localhost`.

-p Specifies the port number that the server uses. Default is 389. If -Z is used, the default is 636.

-w Specifies the password associated with the distinguished name specified in the -D parameter. For example, `-w mypassword`. The default is "", or anonymous.

SSL Parameters

You can use the following parameters to specify that `ldapdelete` use LDAPS when communicating with your Directory Server. You also use these parameters if you want to use certificate-based authentication. These parameters are valid only when LDAPS has been turned on and configured for your Directory Server. For more information on certificate-based authentication,

see “Using Certificate-Based Authentication” on page 311. For information on creating a certificate database for use with LDAP clients, see “Creating Certificate Databases for LDAP Clients” on page 313.

Make sure that you specify your Directory Server’s encrypted port when you use these parameters:

-I FORTEZZA Only. Specifies the personal identification number (PIN) associated with the FORTEZZA crypto card and certificate you specified in the -Q parameter.

-K Specifies the name of the certificate key used for certificate-based client authentication. For example, `-K Server-Key`.

-N Specifies the certificate name to use for certificate-based client authentication. For example, `-N Server-Cert`. If this parameter is specified, then the -Z, -K, and -W parameters are required. Also, if this parameter is specified, then the -D and -w parameters must not be specified, or certificate-based authentication will not occur and the bind operation will use the authentication credentials specified on -D and -w.

-P Specifies the path and filename of the security files for the client. This parameter is used only with the -Z parameter. When used on a machine where an SSL-enabled web browser is configured, the path specified on this parameter can point to the security database for the web browser. For example, `-P c:\security\cert.db`. The client security files can also be stored on the Directory Server in the `<`*NSHOME*`>/alias` directory (where *NSHOME* is the directory where you installed the server). In this case, the -P parameter calls out a path and filename similar to the following: `-P c:\Netscape\Suitespot\alias\client-cert.db`.

-Q FORTEZZA Only. Specifies the number of the slot into which you plugged your FORTEZZA crypto card and, optionally, the name of the FORTEZZA certificate you want to use. The slot number and certificate name are separated by a colon. For example, if you plugged your crypto card into slot 2 and want to use the certificate named `doe`, you would specify the following: `-Q 2:doe`.

-W Specifies the password for the certificate database identified on the -P parameter. For example, `-W serverpassword`.

-X FORTEZZA Only. Specifies the path and filename of the compromised key list (CKL).

-Z Specifies that SSL is to be used for the delete request.

Additional ldapdelete Parameters

The following parameters offer additional functionality:

-c Specifies that the utility run in continuous operation mode. Errors are reported, but the utility continues with deletions. The default is to quit after reporting an error.

-f Specifies the file containing the distinguished names of entries to be deleted. For example, `-f modify_statements`. Omit this parameter if you want to supply the distinguished name of the entry to be deleted directly to the command line.

-H Lists all available `ldapdelete` parameters.

-M Manage smart referrals. Causes the server to not return the smart referral contained on the entry, but to instead delete the actual entry containing the smart referral. For more information about smart referrals, see “Creating and Changing Smart Referrals” on page 360.

-n Specifies that the entries are not to be actually deleted, but that `ldapdelete` is to show what it would do with the specified input.

-O Specifies the maximum number of referral hops to follow. For example, `-O 2`.

-R Specifies that referrals are not to be followed automatically. By default, the server follows referrals.

-v Specifies that the utility is to run in verbose mode.

-V Specifies the LDAP version number to be used on the operation. For example, `-V 2`. LDAP v3 is the default. Note that you cannot perform an LDAP v3 operation against a Directory Server that only supports LDAP v2.

-y Specifies the proxy DN to use for the delete operation. This argument is provided for testing purposes. For more information about proxied authorization, see “Overview of Proxied Authorization” on page 158.

ldapdelete Examples

Suppose:

- You want to delete the entries identified by the distinguished names: `cn=Robert Jenkins, ou=People, o=airius.com` and `cn=Lisa Jangles, ou=People, o=airius.com`.
- You have configured a special entry for the database administrator that has the authority to delete the entries, and that entry has the distinguished name of `cn=Directory Manager, o=airius.com`.
- The database administrator's password is `King~Pin`.
- The Directory Server is located on hostname `cyclops`.
- The Directory Server uses port number 845.

Then to delete the entries for users Robert Jenkins and Lisa Jangles, enter the following command:

```
ldapdelete -D "cn=Directory Manager, o=airius.com" -w King~Pin -h
cyclops -p 845 "cn=Robert Jenkins, ou=People, o=airius.com" "cn=Lisa
Jangles, ou=People, o=airius.com"
```

To delete user Patricia Fuentes from the Airius Bolivia, S.A. tree, you would enter the following command:

```
ldapdelete -D "cn=Directory Manager, o=airius.com" -w King~Pin -h
cyclops -p 845 "cn=Patricia Fuentes, ou=People, o=Airius Bolivia\, S.A."
```

The DN of this entry contains a comma, so you must escape the comma with a backslash (\).

LDIF Update Statements

You use LDIF update statements to define how `ldapmodify` should change your directory. In general, LDIF update statements are a series of statements that:

- Specify the distinguished name of the entry to be modified.
- Specify a changetype that defines how a specific entry is to be modified (add, delete, modify, modrdn, or rename).

  A changetype is required unless you specify the `-a` parameter. If you specify the `-a` parameter, then an add operation (`changetype: add`) is assumed. However, any other changetype overrides the `-a` parameter.

  If you specify a modify operation (`changetype: modify`), a change operation is required that indicates how the entry should be changed.

  If you specify `changetype: modrdn`, change operations are required that specify how the relative distinguished name (RDN) is to be modified. A distinguished name's RDN is the left-most value in the DN. For example, the distinguished name *uid=ssarette, o=airius.com* has an RDN of *uid=ssarette*.
- Specify a series of attributes and their changed values.

The general format of LDIF update statements is as follows:

```
dn: <distinguished name>
<changetype identifier>
<change operation identifier>
<list of attributes>

...

-

<change operation identifier>
<list of attributes>
...

-

...
```

Note A dash (-) must be used to denote the end of a change operation if subsequent change operations are specified. For example, the following statement adds the telephone number and manager attributes to the entry:

```
dn: cn=Lisa Jangles, ou=People, o=airius.com
changetype: modify
add: telephonenumber
telephonenumber: (408) 555-2468
-
add: manager
manager: cn=Harry Cruise, ou=People, o=airius.com
```

In addition, the line continuation operator is a single space. Therefore, the following two statements are identical:

```
dn: cn=Lisa Jangles, ou=People, o=airius.com
```

```
dn: cn=Lisa Jangles,
 ou=People,
 o=airius.com
```

The following sections describe the changetypes in detail.

Adding an Entry Using LDIF

You use `changetype: add` to add an entry to your directory. When you add an entry, make sure to create an entry representing a branch point before you try to create new entries under that branch. That is, if you want to place an entry in a People and an Groups subtree, then create the branch point for those subtrees before creating entries within the subtrees.

The following LDIF update statements can be used to create the People and the Groups subtrees, and then create entries within those trees:

```
dn: o=airius.com
changetype: add
objectclass: top
objectclass: organization
o: airius.com
```

```
dn: ou=People, o=airius.com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: People
ou: Marketing
```

```
dn: cn=Pete Minsky, ou=People, o=airius.com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Pete Minsky
givenName: Pete
sn: Minsky
ou: People
ou: Marketing
uid: pminsky

dn: cn=Sue Jacobs, ou=People, o=airius.com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Sue Jacobs
givenName: Sue
sn: Jacobs
ou: People
ou: Marketing
uid: sjacobs

dn: ou=Groups, o=airius.com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: Groups

dn: cn=Administrators, ou=Groups, o=airius.com
changetype: add
objectclass: top
objectclass: groupOfNames
member: cn=Sue Jacobs, ou=People, o=airius.com
member: cn=Pete Minsky, ou=People, o=airius.com
cn: Administrators

dn: ou=Airius Bolivia\, S.A., o=airius.com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: Airius Bolivia\, S.A.
```

```
dn: cn=Carla Flores, ou=Airius Bolivia\, S.A., o=airius.com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Carla Flores
givenName: Carla
sn: Flores
ou: Airius Bolivia\, S.A.
uid: cflores
```

Using the ldapmodify -a Parameter

If you are simply adding entries to your directory, you can avoid the `changetype: add` statement by specifying the -a parameter on the call to `ldapmodify`. In this case, simply provide LDIF (as opposed to LDIF update statements) to ldapmodify. For example:

```
> ldapmodify -a -h <hostname> -p <port> -D <bind dn> -w <password>
dn: o=airius.com
objectclass: top
objectclass: organization
o: airius.com

dn: ou=People, o=airius.com
objectclass: top
objectclass: organizationalUnit
ou: People

dn: cn=Pete Minsky, ou=People, o=airius.com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Pete Minsky
givenName: Pete
sn: Minsky
ou: People
ou: Marketing

...

^z
```

Renaming an Entry Using LDIF

You use `changetype:modrdn` to change an entry's relative distinguished name (RDN). An entry's RDN is the leftmost element in the distinguished name. Therefore, the RDN for:

```
cn=Barry Nixon, ou=People, o=airius.com
```

is:

```
cn=Barry Nixon
```

And the RDN for:

```
ou=People, o=airius.com
```

is:

```
ou=People
```

Therefore, this rename operation allows you to change the left-most value in an entry's distinguished name. For example, the entry:

```
cn=Sue Jacobs, ou=People, o=airius.com
```

can be modified to be:

```
cn=Susan Jacobs, ou=People, o=airius.com
```

but it cannot be modified to be:

```
cn=Sue Jacobs, ou=old employees, o=airius.com
```

The following example can be used to rename Sue Jacobs to Susan Jacobs:

```
dn: cn=Sue Jacobs, ou=Marketing, o=airius.com
changetype: modrdn
newrdn: cn=Susan Jacobs
deleteoldrdn: 0
```

Because `deleteoldrdn` is `0`, this example retains the existing RDN as a value in the new entry. The resulting entry would therefore have a common name (`cn`) attribute set to both Sue Jacobs and Susan Jacobs in addition to all the other attributes included in the original entry. However, if you used

```
dn: cn=Sue Jacobs, ou=Marketing, o=airius.com
changetype: modrdn
newrdn: cn=Susan Jacobs
deleteoldrdn: 1
```

then the server would delete `cn=Sue Jacobs` and only `cn=Susan Jacobs` would remain within the entry.

A Note on Renaming Entries

You cannot rename an entry with the modrdn changetype such that the entry moves to a completely different subtree. To move an entry to a completely different branch you must create a new entry in the alternative subtree using the old entry's attributes, and then delete the old entry.

Also, for the same reasons that you cannot delete an entry if it is a branch point, you cannot rename an entry if it has any children. Doing so would orphan the children in the tree, which is not allowed by the LDAP protocol. For example, of the following three entries:

```
ou=People, o=airius.com
cn=Paula Simon, ou=People, o=airius.com
cn=Jerry O'Connor, ou=People, o=airius.com
```

you can only rename the last two entries. The entry that identifies the People subtree can only be renamed if no other entries exist below it.

Modifying an Entry Using LDIF

Use `changetype:modify` to add, replace, or remove attributes and/or attribute values to the entry. When you specify `changetype:modify`, you must also provide a change operation to indicate how the entry is to be modified. Change operations can be:

- `add: <attribute>`—Adds the specified attribute or attribute value. If the attribute type does not currently exist for the entry, then the attribute and its corresponding value are created. If the attribute type already exists for the entry, then the specified attribute value is added to the existing value. If the particular attribute value already exists for the entry, then the operation fails and the server returns an error.
- `replace: <attribute>`—The specified values are used to entirely replace the attribute's value(s). If the attribute does not already exist, it is created. If no replacement value is specified for the attribute, the attribute is deleted.
- `delete: <attribute>`—The specified attribute is deleted. If more than one value of an attribute exists for the entry, then all values of the attribute are deleted in the entry. To delete just one of many attribute values, specify the attribute and associated value on the line following the delete change operation.

This section contains the following topics:

Adding Attributes to Existing Entries Using LDIF

You use `changetype:modify` with the add operation to add an attribute and an attribute value to an entry.

For example, the following LDIF update statement adds a telephone number to the entry:

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
add: telephonenumber
telephonenumber: 555-1212
```

The following example adds two telephone numbers to the entry:

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
add: telephonenumber
telephonenumber: 555-1212
telephonenumber: 555-6789
```

The following example adds two `telephonenumber` attributes and a `manager` attribute to the entry:

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
add: telephonenumber
telephonenumber: 555-1212
telephonenumber: 555-6789
-
add: manager
manager: cn=Sally Nixon, ou=People, o=airius.com
```

The following example adds a jpeg photograph to the directory. The jpeg photo can be displayed by the gateway. Note that in order to add this attribute to the directory, you must use the ldapmodify -b parameter (which indicates that ldapmodify should read the referenced file for binary values if the attribute value begins with a slash (/)):

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
add: jpegphoto
jpegphoto: /path/to/photo
```

Changing an Attribute Value Using LDIF

You use `changetype:modify` with the replace operation to change all values of an attribute in an entry.

For example, the following LDIF update statement changes Barney's manager from Sally Nixon to Wally Hensford:

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
replace: manager
manager: cn=Wally Hensford, ou=People, o=airius.com
```

If the entry has multiple instances of the attribute, then to change one of the attribute values, you must delete the attribute value that you want to change, and then add the replacement value. For example, consider the following entry:

```
cn=Barney Fife, ou=People, o=airius.com
objectClass: inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-1212
telephonenumber: 555-5678
```

To change 555-1212 to 555-4321, use the following LDIF update statement:

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
delete: telephonenumber
telephonenumber: 555-1212
-
add: telephonenumber
telephonenumber: 555-4321
```

Barney's entry now is now as follows:

```
cn=Barney Fife, ou=People, o=airius.com
objectClass: inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-5678
telephonenumber: 555-4321
```

Deleting All Values of an Attribute Using LDIF

You use `changetype:modify` with the delete operation to delete an attribute from an entry. If the entry has more than one instance of the attribute, you must indicate which of the attributes you want to delete.

For example, the following LDIF update statement deletes all instances of the `telephonenumber` attribute from the entry, regardless of how many times it appears in the entry:

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
delete: telephonenumber
```

If you want to delete just a specific instance of the `telephonenumber` attribute, then you simply delete that specific attribute value. The following section describes how to do this.

Deleting a Specific Attribute Value Using LDIF

You use `changetype:modify` with the delete operation to delete an attribute value from an entry. You must then indicate which of the actual attributes you want to delete.

For example, consider the following entry:

```
cn=Barney Fife, ou=People, o=airius.com
objectClass: inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-1212
telephonenumber: 555-5678
```

To delete the 555-1212 telephone number from this entry, use the following LDIF update statement:

```
dn: cn=Barney Fife, ou=People, o=airius.com
changetype: modify
delete: telephonenumber
telephonenumber: 555-1212
```

Barney's entry then becomes:

```
cn=Barney Fife, ou=People, o=airius.com
objectClass: inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-5678
```

Deleting an Entry Using LDIF

You use `changetype:delete` to delete an entry from your directory. You can only delete entries at the end of a branch. Therefore, when you delete an entry, make sure that no other entries exist under that entry in the directory tree. That is, you cannot delete an organizational unit entry unless you have first deleted all the entries that belong to the organizational unit.

For example, of the following three entries:

```
ou=People, o=airius.com
cn=Paula Simon, ou=People, o=airius.com
cn=Jerry O'Connor, ou=People, o=airius.com
```

you can delete only the last two entries. The entry that identifies the People subtree can be deleted only if no other entries exist below it.

The following LDIF update statements can be used to delete person entries:

```
dn: cn=Pete Minsky, ou=People, o=airius.com
changetype: delete

dn: cn=Sue Jacobs, ou=People, o=airius.com
changetype: delete
```

Modifying an Entry in an Internationalized Directory

If the attribute values in your directory are associated with one or more languages other than English, the attribute values are associated with language tags. When using the `ldapmodify` command-line utility to modify an attribute that has an associated language tag, you must match the value and language tag exactly or the modify operation will fail.

For example, if you want to modify an attribute value that has a language tag of lang-fr, you must include the lang-fr in the modify operation as follows:

```
dn: bjensen, o=airius.com
changetype: modify
replace: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34 rue Seine
```

Chapter

10

Managing Your Directory Server

This chapter describes basic directory server management. Specifically, this chapter describes:

Viewing and Configuring Log Files

The Netscape Directory Server provides three types of logs to help you better manage your server and tune performance. These logs are described in the following sections:

Access Log

The access log contains detailed information about client connections to the directory.

Viewing the Access Log

To view the access log for the directory server:

1. On the Directory Server Console, select the Status tab and then select the Logs icon in the navigation tree in the left pane.
2. Select the Access Log tab in the right pane.

   This tab displays the last 25 entries in the access log by default.
3. To refresh the current display, click Refresh. Select the Continuous checkbox if you want the display to refresh automatically every ten seconds.
4. To view an archived access log, select it in the Select Log pull-down menu.
5. To display a different number of messages, enter the number you want to view in the Lines to show text box and then click Refresh.
6. You can tell the server to only display messages containing a string you specify. To do this, enter the string in the Show only lines containing text box and then click Refresh.

Configuring the Access Log

You can configure a number of settings to customize the access log, including where the server stores the access log and the creation and deletion policies. You can also disable access logging for the server. You may want to do this because the access log can grow very quickly, (every 2,000 accesses to your server will grow your access log by approximately 1 MB). However, before you turn off access logging, consider that the access log provides beneficial troubleshooting information. To configure the access log for your server:

1. On the Directory Server Console, select the Configuration tab and then select the Logs icon in the navigation tree.

2. Select the Access Log tab in the right pane.

3. To enable access logging, select the Enable Logging checkbox.

   Clear this checkbox if you do not want the server to maintain an access log.

   You can also disable access logging manually by changing the `accesslog-logging-enabled` parameter in the `slapd.conf` file as follows:

   ```
   accesslog-logging-enabled off
   ```

   For information on changing server parameters from `slapd.conf`, see “Changing Parameter Values Using slapd.conf” on page 402.

4. Enter the full path and filename you want the server to use for the access log in the text box provided. The default is:

   ```
   <NSHOME>/slapd-<serverID>/logs/access
   ```

5. If you want the server to periodically archive the current access log and start a new one, define the log file creation policy as follows:

   - Enter the total number of access logs you want the server to keep in the ”Maximum Number of Logs” text box. When the server exceeds this amount, it deletes the oldest archived access log file in the folder. The default is 10 logs. Do not set this value to 1. If you do, the server will not rotate the log and it will grow indefinitely.

- Enter the maximum size (in MB) you want the server to allow access logs to reach in the "File size for each log" text box. If you don't want to set a maximum size, type `-1` in this field. Once a log file reaches this maximum size (or the maximum age defined in the next step), the server archives the file and starts a new one. The default is 100 MB. If you set the Maximum Number of Logs to 1, the server ignores this parameter.

- Define how often you want the server to archive the current access log file and create a new one by entering a number of minutes, hours, days, weeks, or months in the "Create a new log every" text box. The default is every day. If you set the Maximum Number of Logs to 1, the server ignores this parameter.

6. If you want the server to automatically delete old archived access logs, define the log file deletion policy as follows:

   - If you want the server to delete the oldest archived access log once the total size of the combined archived logs reaches a certain amount, enter the maximum size (in MB) in the "When total log size exceeds" text box. If you don't want to set a maximum size, type `-1` in this field. The default is 500 MB.

   - If you want the server to delete the oldest archived access log once the free disk space goes below a certain amount, enter a value (in MB) in the "When free disk space is less than" text box. The default is 5 MB.

   - You can configure the server to delete old access logs according to their age by entering a number of days, weeks, or months in the "When a file is older than" text box. The default is 1 month.

7. When you are finished making changes, click Save.

Error Log

The error log contains detailed messages of errors and events the server experiences during normal operations.

Viewing the Error Log

To view the error log for the directory server:

1. On the Directory Server Console, select the Status tab and then select the Logs icon in the navigation tree.
2. Select the Error Log tab in the right pane.

   This tab displays the last 25 entries in the error log by default.
3. To refresh the current display, click Refresh. Select the Continuous checkbox if you want the display to refresh automatically every ten seconds.
4. To view an archived error log, select it in the Select Log pull-down menu.
5. To specify a different number of messages, enter the number you want to view in the Lines to show text box and click Refresh.
6. You can tell the server to only display messages containing a string you specify. To do this, enter the string in the Show only lines containing text box and click Refresh.

Configuring the Error Log

You can change several settings for the error log, including where the server stores the log and what you want the server to include in the log. To configure the error log, complete the following:

1. On the Directory Server Console, select the Configuration tab and then select the Logs icon in the navigation tree.
2. Select the Error Log tab in the right pane.

3. To enable error logging, select the Enable Logging checkbox.

   Clear this checkbox if you do not want the server to maintain an error log.

   You can also disable error logging manually by changing the `errorlog-logging-enabled` parameter in the `slapd.conf` file as follows:

   ```
   errorlog-logging-enabled off
   ```

   For information on changing server parameters from `slapd.conf`, see "Changing Parameter Values Using slapd.conf" on page 402.

4. Enter the full path and filename you want the server to use for the error log in the text box provided. The default is:

   ```
   <NSHOME>/slapd-<serverID>/logs/error
   ```

5. If you want the server to periodically archive the current error log and start a new one, define the log file creation policy as follows:

   - Enter the total number of error logs you want the server to keep in the "Maximum Number of Logs" text box. When the server exceeds this amount, it deletes the oldest archived error log file in the folder. The default is 1 log. If you accept this default, the server will not rotate the log and it will grow indefinitely.

   - Enter the maximum size (in MB) you want the server to allow error logs to reach in the "File size for each log" text box. If you don't want to set a maximum size, type `-1` in this field. Once a log file reaches this maximum size (or the maximum age defined in the next step), the server archives the file and starts a new one. The default is 100 MB. If you set the Maximum Number of Logs to 1, the server ignores this parameter.

   - Define how often you want the server to archive the current error log file and create a new one by entering a number of minutes, hours, days, weeks, or months in the "Create a new log every" text box. The default is every week. If you set the Maximum Number of Logs to 1, the server ignores this parameter.

6. If you want the server to automatically delete old archived error logs, define the log file deletion policy as follows:

   - If you want the server to delete the oldest archived error log once the total size of the combined archived logs reaches a certain amount, enter the maximum size (in MB) in the "When total log size exceeds" text box. If you don't want to set a maximum size, type `-1` in this field. The default is 100 MB.

   - If you want the server to delete the oldest archived error log once the free disk space goes below a certain amount, enter a value (in MB) in the "When free disk space is less than" text box. The default is 5 MB.

   - You can configure the server to delete old error logs based on the log's age by entering a number of days, weeks, or months in the "When a file is older than" text box. The default is 1 month.

7. If you want to set the log level, Ctrl+click the options you want the server to include in the Log Level list box. For more information about log level options, see "Log Level" on page 444.

   Changing these values from the defaults may cause your error log to grow very rapidly, so it is recommended that you do not change your logging level unless you are asked to by Netscape Customer Support.

8. When you are finished making changes, click Save.

Audit Log

The audit log contains detailed information about changes made to each database as well as to server configuration.

Viewing the Audit Log

Before you can view the audit log, you must enable audit logging for the server. See “Configuring the Audit Log” on page 270 for information. To view the audit log for the directory server, complete the following:

1. On the Directory Server Console, select the Status tab and then select the Logs icon in the navigation tree.
2. Select the Audit Log tab in the right pane.

   This tab displays the last 25 entries in the audit log by default.
3. To refresh the current display, click Refresh. Select the Continuous checkbox if you want the display to refresh automatically every ten seconds.
4. To view an archived audit log, select it in the Select Log pull-down menu.
5. To display a different number of messages, enter the number you want to view in the Lines to show text box and click Refresh.
6. You can tell the server to only display messages containing a string you specify. To do this, enter the string in the Show only lines containing text box and click Refresh.

Configuring the Audit Log

You can use the Directory Server Console to enable and disable audit logging and to specify where the audit log file is stored. To configure audit logging, complete the following:

1. On the Directory Server Console, select the Configuration tab and then select the Logs icon in the navigation tree.
2. Select the Audit Log tab in the right pane.
3. To enable audit logging, select the Enable Logging checkbox.

   To disable audit logging, clear the checkbox. By default, audit logging is disabled.

You can also disable audit logging manually by changing the `auditlog-logging-enabled` parameter in the `slapd.conf` file as follows:

```
auditlog-logging-enabled off
```

For information on changing server parameters from `slapd.conf`, see “Changing Parameter Values Using slapd.conf” on page 402.

4. Enter the full path and filename you want the server to use for the audit log in the text box provided. The default is:

   ```
   <NSHOME>/slapd-<serverID>/logs/audit
   ```

5. If you want the server to periodically archive the current audit log and start a new one, define the log file creation policy as follows:

   - Enter the total number of audit logs you want the server to keep in the ”Maximum Number of Logs” text box. When the server exceeds this amount, it deletes the oldest archived audit log file in the folder.The default is 1 log. If you accept this default, the server will not rotate the log and it will grow indefinitely.

   - Enter the maximum size (in MB) you want the server to allow audit logs to reach in the “File size for each log” text box. If you don’t want to set a maximum size, type `-1` in this field. Once a log file reaches this maximum size (or the maximum age defined in the next step), the server archives the file and starts a new one. If you set the Maximum Number of Logs to 1, the server ignores this parameter.

   - Define how often you want the server to archive the current audit log file and create a new one by entering a number of minutes, hours, days, weeks, or months in the “Create a new log every” text box. If you set the Maximum Number of Logs to 1, the server ignores this parameter.

6. If you want the server to automatically delete old archived audit logs, define the log file deletion policy as follows:

   - If you want the server to delete the oldest archived audit log once the total size of the combined archived logs reaches a certain amount, enter the maximum size (in MB) in the “When total log size exceeds” text box.
   - If you want the server to delete the oldest archived audit log once the free disk space goes below a certain amount, enter a value (in MB) in the “When free disk space is less than” text box.
   - You can configure the server to delete old audit logs based on the log’s age by entering a number of days, weeks, or months in the “When a file is older than” text box.

7. When you are finished making changes, click Save.

Manual Log File Rotation

The directory server supports automatic log file rotation for all three logs. However, you can manually rotate log files if you have not set automatic log file creation or deletion policies. By default, access, error, and audit log files can be found in the following location:

```
<NSHOME>/slapd-<serverID>/logs/
```

To manually rotate log files, do the following:

1. Shut down the server. See “Starting and Stopping the Directory Server” on page 29 for more information.
2. Move or rename the log file you are rotating. You might want to keep the old log file for future reference.
3. Restart the server.

Monitoring Server Activity

You can monitor your directory server's current activities from either the server console or the command line. For information on how to monitor your server's activity from the command line, refer to "Monitoring Your Server From the Command Line" on page 277.

Monitoring Your Server From the Server Console

To monitor your server's activities through the server console:

1. On the Directory Server Console, select the Status tab and click Performance Counters in the navigation tree in the left pane.

   The Server tab in the right pane displays current information about server activity. If the server is currently not running, this tab will not provide performance monitoring information.

2. Click Refresh to refresh the currently displayed information. If you want the server to continuously update the displayed information, select the Continuous checkbox.

The server provides server monitoring information as described in the following sections:

- "General Information (Server)" on page 274
- "Resource Summary" on page 275
- "Current Resource Usage" on page 276
- "Connection Status" on page 277

General Information (Server)

The server provides the following general information:

- Server version—identifies the current server version.
- Config DN—identifies the server's machine data tree DN. For information on machine data, see "Machine data" on page 356.
- Data version—provides identification information for the server's data area. Usually the information shown here is only relevant if your server is supplying replicated trees to consumer servers. The data version information is supplied as follows:
  - Server host name.
  - Server port number.
  - Database generation number; a unique identifier that is created only when you create your directory database without a machine data entry in the LDIF file.
  - Current change log number. This is the number corresponding to the last change made to your directory. This number starts at one and increments by one for each change made to the database.
- Startup time on server—Date and time the server was started.
- Current time on server—Displays the current date and time on the server.

Resource Summary

The Resource Summary table provides the resource-specific information described in Table 10.1.

Table 10.1 Server Performance Monitoring - Resource Summary table

Resource	Usage since startup	Average per minute
Connections	Total number of connections to this server since server startup.	Average number of connections per minute since server startup.
Operations Initiated	Total number of operations initiated since server startup. Operations include any client requests for server action, such as searches, adds, and modifies in the directory tree. It is likely that multiple operations will be initiated for each connection.	Average number of operations per minute since server startup.
Operations Completed	Total number of operations completed by the server since server startup.	Average number of operations per minute since server startup.
Entries sent to clients	Total number of entries sent to clients since server startup. Entries are sent to clients as the result of search requests.	Average number of entries sent to clients per minute since server startup.
Bytes sent to clients	Total number of bytes sent to clients since server startup.	Average number of bytes sent to clients per minute since server startup.

Current Resource Usage

The Resource Summary table provides the resource-specific information described in Table 10.2.

Table 10.2 Server Performance Monitoring - Current Resource Usage table

Resource	Current total
Active Threads	Current number of active threads used for handling requests. Additional threads may also be created by internal server tasks, such as replication.
Open Connections	Total number of open connections. Each connection can account for multiple operations, and therefore multiple threads.
Remaining available connections	Total number of remaining connections that the server can concurrently open. This number is based on the number of currently open connections, and the total number of concurrent connections that the server is allowed to open. In most cases, the latter value is determined by the operating system, and is expressed as the number of file descriptors available to a task. On Windows NT and IBM AIX, the number is generated by the operating system, but is not based on file descriptors. Refer to your operating system documentation for more information.
Threads waiting to write to client	Total number of threads waiting to write to the client. This happens anytime the server must pause while sending data to a client. Reasons for this may include a slow network or client, or an extremely large amount of information being sent to the client.
Threads waiting to read from client	Total number of threads waiting to read from the client. This happens if the server starts to receive a request from the client and then the transmission of that request is halted for some reason. Generally, threads waiting to read are an indication of a slow network or client.
Thread Concurrency	Meaningful on Solaris 2.x only. Provides an indication of the level of thread concurrency.
Databases in use	Total number of databases being serviced by the server. Currently, this value is always 1.

Connection Status

The Connection Status table provides information on the amount of resources in use by each currently open connection as described in Table 10.3.

Table 10.3 Server Performance Monitoring - Connection Status table

Table Header	Description
Time opened	Indicates the time on the server when the connection was initially opened.
Started	Indicates the number of operations initiated by this connection.
Completed	Indicates the number of operations completed by the server for this connection.
Bound as	Indicates the distinguished name used by the client to connect to the server. If the client has not authenticated to the server, the server displays `not bound` in this field.
Read/Write	Indicates whether the server is currently blocked for read or write access to the client. Possible values include: • Not blocked—Indicates that the server is idle, actively sending data to the client, or actively reading data from the client. • Blocked—Indicates that the server is trying to send data to the client or read data from the client, but cannot. The probable cause is a slow network or client.

Monitoring Your Server From the Command Line

You can monitor your directory server's current activities from any LDAP client by performing a search against:

```
objectClass=*
```

and a search base of:

```
cn=monitor
```

and a scope of:

```
base
```

For example:

```
ldapsearch -h directory.airius.com -s base
-b “cn=monitor” "(objectclass=*)"
```

For information on searching the directory server, see “Using ldapsearch” on page 212.

When you monitor your server's activities in this way, you see the following information:

`version:`

Identifies the directory server's current version number.

`threads:`

Current number of active threads used for handling requests. Additional threads may also be created by internal server tasks, such as replication, or writing to logs.

`connection: <fd>:<opentime>:<opsinitiated>:<opscompleted>:<binddn>:[rw]`

Provides the following summary information for each open connection (only available if you bind to the directory as the Root DN):

- `fd`—The file descriptor used for this connection.
- `opentime`—The time this connection was opened.
- `opsinitiated`—The number of operations initiated by this connection.
- `opscompleted`—The number of operations completed.
- `binddn`—The distinguished name used by this connection to connect to the directory server.
- `rw`—Field that is shown if the connection is blocked for read or write.

`currentconnections:`

Identifies the number of connections currently in service by the directory server.

`totalconnections:`

Identifies the number of connections handled by the directory server since it started.

`dtablesize:`

Shows the number of file descriptors available to the directory server. Each connection requires one file descriptor; one for every open index, one for log file management, and one for ns-slapd itself. Essentially, this value lets you know about how many more concurrent connections can be serviced by the directory server.

For more information on file descriptors, refer to your operating system documentation.

`writewaiters:`

Identifies the number of threads waiting to write data to a client.

`readwaiters:`

Identifies the number of threads waiting to read data from a client.

`opsinitiated:`

Identifies the number of operations the server has initiated since it started.

`opscompleted:`

Identifies the number of operations the server has completed since it started.

`entriessent:`

Identifies the number of entries sent to clients since the server started.

`bytessent:`

Identifies the number of bytes sent to clients since the server started.

`currentime:`

Identifies the time when this snapshot of the server was taken. The time is displayed in Greenwich mean time (GMT) in UTC format.

`starttime:`

Identifies the time when the server started. The time is displayed in Greenwich mean time (GMT) in UTC format.

`nbackends:`

Identifies the number of back ends (databases) the server services. Currently this value is always one.

`concurrency:`

Solaris 2.x only. Indicates the current level of thread concurrency.

Monitoring Database Activity

You can monitor your database's current activities from the server console or from the command line. For information on how to monitor your database's activities from the command line, refer to "Monitoring the Database From the Command-Line" on page 286.

Monitoring Database Activity From the Server Console

To monitor your database's activities through the server console:

1. On the Directory Server Console, select the Status tab.
2. Select Performance Counters in the navigation tree in the left pane and then select the Database tab in the right pane.

   The Database tab displays current information about database activity. If the server is currently not running, this tab will not provide performance monitoring information.
3. Click Refresh to refresh the currently displayed information. If you want the server to continuously update the displayed information, select the Continuous checkbox and then click Refresh.

The server provides database monitoring information as described in the following sections:

- "General Information (Database)" on page 281
- "Summary Information Table" on page 281

- “Database Cache Information Table” on page 283
- “Database File-Specific Table” on page 285

General Information (Database)

The server provides the following general database information:

- Database—identifies the type of database that you are monitoring.
- Config DN—identifies the distinguished name that you can use to obtain these results using the `ldapsearch` command-line utility.

Summary Information Table

The Summary Information table provides information as described in Table 10.4.

Table 10.4 Database Performance Monitoring - Summary Information table

Performance Metric	Current Total
Readonly status	Indicates whether the database is currently in read-only mode. Your database is in read-only mode when your `readonly slapd.conf` parameter is set to `on`.
Entry cache hits	Indicates the total number of successful entry cache lookups. That is, the total number of times the server could process a search request by obtaining data from the cache rather than by going to disk.
Entry cache tries	Indicates the total number of entry cache lookups since the directory server was last started. That is, the total number of search operations performed against your server since server startup.

Table 10.4 Database Performance Monitoring - Summary Information table (Continued)

Performance Metric	Current Total
Entry cache hit Ratio	Ratio that indicates the number of entry cache tries to successful entry cache lookups. This number is based on the total lookups and hits since the server was last started. The closer this value is to 100% the better. Whenever a search operation attempts to find an entry that is not resident in the entry cache, the directory server has to perform a disk access to obtain the entry. Thus, as this ratio drops towards zero, the number of disk accesses increases and directory server search performance drops. To improve this ratio, you can increase the number of entries that the directory server maintains in the entry cache by increasing the value on the Maximum Entries in Cache parameter in `slapd.ldbm.conf`. See "Tuning Database Performance" on page 290 for information on changing this value using the server console. The maximum value that you can set on this parameter depends on the amount of real memory on your machine as well as the value set for the Maximum Cache Size parameter. That is, `(Maximum Entries in Cache + Maximum Cache Size) x average entry size` should never be greater than the amount of available memory on your machine. Use caution when changing either of these two parameters. Your ability to improve server performance with these parameters depends on the size of your database, the amount of physical memory available on your machine, and whether directory searches are random. If your database will not fit into memory, and if searches are random (that is, if your directory clients are searching for random and widely scattered directory data), attempting to increase the values set on these parameters will not help directory performance, and may in fact harm overall performance.

Table 10.4 Database Performance Monitoring - Summary Information table (Continued)

Performance Metric	Current Total
Current number of entries in entry cache	Indicates the total number of directory entries currently resident in the entry cache.
Maximum number of entries in entry cache	Indicates the maximum number of directory entries that are allowed to be maintained in the entry cache. This value is managed by the `Maximum Entries in Cache` parameter in `slapd.ldbm.conf`. See "Tuning Database Performance" on page 290 for information on changing this value using the server console.

Database Cache Information Table

The Database Cache Information table provides the caching information as described in Table 10.5.

Table 10.5 Database Performance Monitoring - Database Cache Information table

Performance Metric	Current Total
Hits	Indicates the number of times the database cache successfully supplied a requested page.
Tries	Indicates the number of times the database cache was asked for a page.

Table 10.5 Database Performance Monitoring - Database Cache Information table

Performance Metric	Current Total
Hit ratio	Indicates the ratio of database cache hits to database cache tries. The closer this value is to 100%, the better. Whenever a directory operation attempts to find a portion of the database that is not resident in the database cache, the directory server has to perform a disk access to obtain the appropriate database page. Thus, as this ratio drops towards zero, the number of disk accesses increases and directory server performance drops. To improve this ratio, you can increase the amount of data that the directory server maintains in the database cache by increasing the value on the `Maximum Cache Size` parameter in `slapd.ldbm.conf`. See "Tuning Database Performance" on page 290 for information on changing this value using the server console. The maximum value that you can set on this parameter depends on the amount of real memory on your machine as well as the value set for the `Maximum Entries in Cache` parameter. That is, `(Maximum Entries in Cache + Maximum Cache Size) x average entry size` should never be greater than the amount of available memory on your machine. Use caution when changing either of these two parameters. Your ability to improve server performance with these parameters depends on the size of your database, the amount of physical memory available on your machine, and whether directory searches are random. If your database will not fit into memory, and if searches are random (that is, if your directory clients are searching for random and widely scattered directory data), attempting to increase the values set on these parameters will not help directory performance, and may in fact harm overall performance.
Pages read in	Indicates the number of pages read from disk into the database cache.
Pages written out	Indicates the number of pages written from the cache back to disk. A database page is written out to disk whenever a read-write page has been modified and then subsequently evicted from the cache. Pages are evicted from the database cache when the cache is full and a directory operation requires a database page that is not currently stored in cache.

Table 10.5 Database Performance Monitoring - Database Cache Information table

Performance Metric	Current Total
Read-only page evicts	Indicates the number of read-only pages discarded from the cache to make room for new pages.
Read-write page evicts	Indicates the number of read-write pages discarded from the cache to make room for new pages. This value differs from Pages Written Out in that these are discarded read-write pages that have not been modified.

Database File-Specific Table

The server displays a table for each index file that makes up your database. Each of the tables provides the information described in Table 10.6.

Table 10.6 Database Performance Monitoring - Database File-Specific table

Performance Metric	Current Total
Cache hits	Number of times that a search result resulted in a cache hit on this specific file. That is, a search that required data from this file was performed and the required data was successfully obtained from the cache.
Cache misses	Number of times that a search result failed to result in a cache hit on this specific file. That is, a search that required data from this file was performed and the required data could not be found in the cache.
Pages read in	Indicates the number of pages brought to the cache from this file.
Pages written out	Indicates the number of pages for this file written from cache to disk.

Monitoring the Database From the Command-Line

You can monitor your directory server's database activities from any LDAP client by performing a search against

```
objectClass=*
```

and a search base of:

```
cn=monitor,cn=ldbm
```

and a scope of

```
base
```

For example:

```
ldapsearch -h directory.airius.com -s base
 -b “cn=monitor,cn=ldbm” (objectclass=*)
```

For information on searching the directory server, see “Using ldapsearch” on page 212.

When you monitor your server's activities in this way, you see the following information:

`database`

Identifies the type of database you are currently monitoring.

`readonly`

Indicates whether the database is in read-only mode. `0` indicates that the server is not in read-only mode, `1` indicates that it is in read-only mode.

`entrycachehits`

Provides the same information as described in “Entry cache hits” on page 281 in Table 10.4.

`entrycachetries`

Provides the same information as described in “Entry cache tries” on page 281 in Table 10.4.

`entrycachehitratio`

Provides the same information as described in “Entry cache hit Ratio” on page 282 in Table 10.4.

`currententrycachesize`

Provides the same information as described in “Current number of entries in entry cache” on page 283 in Table 10.4.

`maxentrycachesize`

Provides the same information as described in “Maximum number of entries in entry cache” on page 283 in Table 10.4.

`dbchehits`

Provides the same information as described in “Hits” on page 283 in Table 10.5.

`dbcachetries`

Provides the same information as described in “Tries” on page 283 in Table 10.5.

`dbcachehitratio`

Provides the same information as described in “Hit ratio” on page 284 in Table 10.5.

`dbcachepagein`

Provides the same information as described in “Pages read in” on page 284 in Table 10.5.

`dbcachepageout`

Provides the same information as described in “Pages written out” on page 284 in Table 10.5.

`dbcacheroevict`

Provides the same information as described in “Read-only page evicts” on page 285 in Table 10.5.

`dbcacherwevict`

Provides the same information as described in “Read-write page evicts” on page 285 in Table 10.5.

Next the following information for each file that makes up your database is displayed:

`dbfilename-<number>`

Indicates the name of the file. `<number>` provides a sequential integer identifier (starting at 0) for the file. All associated statistics for the file are given this same numerical identifier.

`dbfilecachehit-<number>`

Provides the same information as described in “Cache hits” on page 285 in Table 10.6.

`dbfilecachemiss-<number>`

Provides the same information as described in “Cache misses” on page 285 in Table 10.6.

`dbfilepagein-<number>`

Provides the same information as described in “Pages read in” on page 285 in Table 10.6.

`dbfilepageout-<number>`

Provides the same information as described in “Pages written out” on page 285 in Table 10.6.

Managing the Root DN

The Root DN is the privileged database user; that is, access control does not apply to this user. You initially defined the Root DN during installation. The default is cn=Directory Manager.

The password for this user is defined in the Root Password parameter in `slapd.conf`.

To set your root DN and password and the encryption scheme used for this password:

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
2. Select the Manager tab in the right pane.

3. Enter the new distinguished name for the Root DN in the text box provided.
4. From the “Manager Password Encryption” pull-down menu, select the storage scheme you want the server to use to store the Root DN password.
5. Click Save.

Tuning Performance

There are several parameters available to you that allow you to manage performance. These parameters are described in the following sections:

- “Tuning Server Performance” on page 289
- “Tuning Database Performance” on page 290

Tuning Server Performance

The server parameters let you manage your server’s performance by limiting the amount of resources the server puts into client search requests. LDAP clients can cause the server to actually use smaller values for Size Limit and Time Limit. To configure the server parameters to optimize performance:

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.

   The server-wide configuration tabs appear in the right pane.
2. Select the Performance tab in the right pane.

   The current server performance settings appear.
3. Set the maximum number of entries the server will return to the client in response to a search operation by entering a new value in the “Size Limit” text box.

   If you do not want to set a limit, type `-1` in this text box.

4. Enter the maximum amount of real time (in seconds) you want the server to spend performing a search request in the ”Time Limit” text box.

   If you do not want to set a limit, type zero (`0`) in this text box.

5. Enter the time (in seconds) you want the server to maintain an idle connection before terminating it, in the ”Idle Timeout” text box.

   If you do not want to set a limit, type zero (`0`) in this text box.

6. Set the maximum number of file descriptors available to the directory server in the Max Number of File Descriptors text box.

   This option is not available for Windows NT or IBM AIX. For more information on this parameter, see “Maximum File Descriptors” on page 447.

For a better understanding of how these parameters impact your server’s searching performance, refer to “The Searching Algorithm” on page 176.

Tuning Database Performance

The database parameters influence server performance primarily on searches by defining the amount of memory available to the server. To configure the database parameters to optimize performance:

1. On the Directory Server Console, select the Configuration tab and then select Database in the left pane.

   This displays the Database tabs in the right pane.

2. Select the Performance tab in the right pane.

   The current database performance settings appear.

3. Enter the number of entries you want the server to keep in memory in the “Maximum Entries in Cache” text box.

4. Enter the amount of memory you want to make available for open index files in the “Maximum Cache Size” text box.

   Indexes and index files are described in Chapter 7, “Managing Indexes.” For more information on this parameter, see “Maximum Cache Size” on page 482.

   If you are creating a very large database from LDIF, set this parameter as large as possible. The larger this parameter, the faster your database will be created. As a rule, determine how much free memory you have on your system, divide that number by two, reduce this number by about 1 MB, and set that number on this parameter. For example, if you have 50 MB of free memory on your system, divide by 2 (25 MB) and reduce by 1 MB (24 MB). Set your Maximum cache size in bytes parameter to 24 MB.

   When you are done creating your database, be sure to set this parameter back to some lower value before you run your server in a production environment.

5. Enter the maximum number of entries you want the server to check in response to a search request in the “Look Through Limit” text box.

   If you do not want to set a limit, type -1 in this text box. If you bind to the directory as the Root DN, unlimited is set by default and overrides any settings you specify here.

Managing Network and LDAP Settings

You can view and change the parameters relevant to the server's network and LDAP settings through the Directory Server Console. This section provides information in the following sections:

For information on schema checking, see Chapter 3, “Extending the Directory Schema.”

Changing Directory Server Port Numbers

You can modify the port or secure port number of your user directory server using the directory server console or by changing the value in slapd.conf. See “Port Number” on page 459 for more information.

If you want to modify the port or secure port for a Netscape Directory Server that contains the Netscape configuration information (`o=NetscapeRoot` subtree), you may do so through the Directory Server Console, or by changing the value in both `slapd.conf` and in the corresponding SIE in the configuration directory.

If you change the configuration directory or user directory port or secure port numbers, you should be aware of the following repercussions:

- You need to change the configuration or user directory port or secure port number configured for the Administration Server. See *Managing Servers with Netscape Console* for information.
- If you have other Netscape Servers installed that point to the configuration or user directory, you need to update those servers to point to the new port number.

To modify the port or secure port on which either a user or a configuration directory listens for incoming requests:

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
2. Select the Settings tab in the right pane.
3. Enter the port number you want the server to use for non-SSL communications in the “Port” text box.

4. Enter the port number you want the server to use for SSL communications in the Encrypted Port text box.

   The encrypted port number that you specify must not be the same port number as you are using for normal LDAP communications.

5. Click Save and then restart the server. See “Starting and Stopping the Directory Server” on page 29 for more information.

Enabling the Directory Server to use the NT Synchronization Service

The NT Synchronization Service causes the directory server to start verifying changes made to NT user and group information, and to transmit changes made to NT user and group information to the NT Primary Domain Controller (PDC). Also, the NT Synchronization Service propogates changes made to user and group information from the PDC to the directory server. For information on how directory server to PDC synchronization occurs, see “How Synchronization Occurs” on page 370. For more information about using the NT Synchronization Service, see Chapter 15, “NT Directory Synchronization.”

To enable the Directory Server to use the NT Synchronization Service:

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.

2. Select the Settings tab in the right pane.

3. Select the “Enable NT Synchronization Service” checkbox.

4. Select the “Use SSL in NT Synchronization Service” checkbox if you want to configure the Directory Server and the synchronization service to use SSL during communications.

5. Specify a Synchronization Port Number. The NT Synchronization Service negotiates changes initiated from the directory server using this port.

6. Click Save and then restart the server. See “Starting and Stopping the Directory Server” on page 29 for more information.

Placing the Entire Directory Server in Read-only Mode

If you maintain more than one database with your directory server and you need to place all your databases in read-only mode, you can place each database in read-only mode individually, or you can place them all in read-only mode at the same time using this option:

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
2. Select the Settings tab in the right pane.
3. Select the “Make entire server read-only” checkbox.
4. Click Save and then restart the server.

Tracking Modifications to Directory Entries

You can configure the server to maintain special attributes for newly created or modified entries. If you are using your directory server with the NT Synchronization Service, then you must select this option. By default, the track modifications option is enabled.

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
2. Select the Settings tab in the right pane.

3. Select the “Track Entry Modification Times” checkbox.

   The server adds the following attributes to a newly created or modified entry:

   - `modifiersName`—The distinguished name of the person who last modified the entry.
   - `modifyTimestamp`—The timestamp for when the entry was last modified in GMT format.
   - `creatorsName`—The distinguished name of the person who initially created the entry.
   - `createTimestamp`—The timestamp for when the entry was created in GMT format.

4. Click Save and then restart the server. See “Starting and Stopping the Directory Server” on page 29 for more information.

Chapter

11

Managing SSL

To provide secure communications over the network, the Netscape Directory Server provides the LDAPS communications protocol. LDAPS is the standard LDAP protocol, but it runs on top of Secure Sockets Layer (SSL).

To use LDAPS, you:

1. Obtain and install a certificate for your directory server, and configure the directory server to trust the certification authority's certificate. For information, see "Obtaining and Installing Server Certificates" on page 298.
2. Turn on SSL in your directory. For information, see "Activating SSL" on page 307.
3. Configure the administration server to connect to an SSL-enabled directory server. For information, see *Managing Servers with Netscape Console*.

If you are using FORTEZZA, please read Chapter 12, "Managing FORTEZZA," for information before you attempt to set up SSL.

For a complete description of SSL, internet security, certificates, and setting up certificate databases, see *Managing Servers with Netscape Console*.

The directory server is capable of simultaneous SSL and non-SSL communications. This means that you do not have to choose between SSL or non-SSL communications for your directory server; you can use both at the same time.

This chapter describes how to use SSL with your directory server in the following sections:

Obtaining and Installing Server Certificates

This section describes the process of creating a certificate database, obtaining and installing a certificate for use with your directory server, and configuring the directory server to trust the certification authority's (CA) certificate. This process is a necessary first step before you can turn on SSL in your directory. If you have already completed these tasks, see "Activating SSL" on page 307. If you are using FORTEZZA with your directory server, see Chapter 12, "Managing FORTEZZA." Obtaining and installing certificates consists of the following five steps:

- Step 1: Generate a Certificate Request
- Step 2: Send the Certificate Request
- Step 3: Install the Certificate
- Step 4: Trust the Certificate Authority
- Step 5: Confirm That Your New Certificates Are Installed

You use the Certificate Setup Wizard to request a certificate from a Certificate Authority, when you are ready to install the certificate, and again to trust the CAs certificate. The Certificate Setup Wizard automates the process of creating and installing the key-pair and certificate database for you. For a complete overview of the Certificate Setup Wizard, see the online help or *Managing Servers with Netscape Console.*

Step 1: Generate a Certificate Request

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

1. On the Directory Server Console, select the Tasks tab and click Certificate Setup Wizard.

   The following dialog box appears outlining the steps required to set up a server certificate. Click Next.

2. On the dialog box that appears, select Internal (software) from the “Select a token (Cryptographic Device)” drop-down menu.

3. Under “Is the server certificate already requested and ready to install?”, choose No if you have never submitted a request for this certificate.

   You would choose Yes when you are ready to install the certificate as described in “Step 3: Install the Certificate” on page 304. You only choose the third option if you are using FORTEZZA. (If you are using FORTEZZA, your key is stored in an external device.) See Chapter 12, “Managing FORTEZZA,” if you are using FORTEZZA.

4. Click Next. If you have already set up a certificate database for the server's host, skip to the next step. If a certificate database does not already exist for this host, click Next again to create one. A certificate database is a key-pair and certificate database installed on the local host. When you use an internal token, the certificate database is the database into which you install the key and certificate.

   On the dialog box that appears, enter and confirm the password you want to use for the certificate database and click Next. The password must contain at least 8 characters, at least one of them numeric. This password helps secure access to the new key database you are creating.

   Once the certificate database is created, the wizard displays a confirmation dialog. Click Next to continue.

5. A dialog appears confirming that the wizard is ready to continue with the certificate setup and indicates that you need to determine the distinguished name for the server and have the information readily available. See the online help for more information. Click Next.

6. The Generating a Certificate Request - Step 1 dialog box appears. If prompted, select a token from the list of legal key tokens you can use, enter the password you used when you set up the certificate database, and then click Next.

7. The Generating a Certificate Request - Step 2 dialog box appears. Select whether this is a request for a new server certificate or whether you are renewing an existing server certificate. If you want to create a new certificate, choose New Certificate. If you already have an existing certificate, the Certificate Renewal option takes less time. If you have an existing certificate and want to replace or renew it, choose Certificate Renewal.

8. Enter the CA administrator's address where your certificate request should be sent. If you want, click Show CA to launch a web browser and view a list of the Certificate Authorities available to you.

9. Click Next. The Generate a Certificate Request - Step 3 dialog box appears. Enter the following information and click Next.

   Your name. Enter your user ID.

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

   Server Host Name. Enter the fully qualified hostname of the directory server as it is used in DNS lookups, for example, `dir.airius.com`.

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

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

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

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

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

   Country. Select the two-character abbreviation for your country's name (ISO format). The country code for the United States is US. The *Netscape Directory Server Schema Reference Guide* contains a complete list of ISO Country Codes.

10. The Generate a Certificate Request - Step 4 dialog box appears. This dialog box contains the certificate request that you need to send to the CA. Click Cancel to exit the wizard.

Once you have generated the request, you are ready to send it to the CA as described in "Step 2: Send the Certificate Request" on page 302.

Step 2: Send the Certificate Request

If you are using Unix, the certificate request is sent for you automatically via sendmail.

If you are using Windows NT, the certificate information is automatically generated and saved to a file under the server host's `\temp` directory. Follow these steps to send the certificate information to the CA:

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

   The file will look similar to the following example:

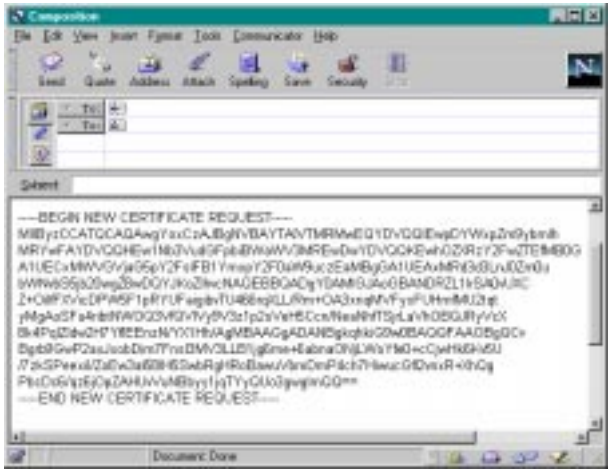
```
Certificate request has been generated.

The mail that you should send is in the file c:\temp\mailtmp.1

It contains the To, Subject and Reply-To fields. Please use your mailer
to enter the rest of the file as the body of the message. When the
response arrives, you can use the Install a Certificate form to put it
in place.


To: ca@airius.com

Subject: Certificate request

Reply-To: bjensen@airius.com


Webmaster: ca@airius.com

Phone: 888 555.1234


Common-name: dirserver.airius.com

Email: bjensen@airius.com

Organization: Airius Corporation

State: CALIFORNIA

Country: Us
```

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UEBhMCVXMxEzARBgNVBAgTCkNBTElGT1JOSUExLDAqBgN
VBAoTI25ldHNjYXBlIGNvbW11bmljYXRpb25zIGNvcnBvcmF0aW9uMRwwGgYDVQQDExNtZW
xsb24ubmV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCwAbskGh6SK
YOgHy+UCSLnm3ok3X3u83Us7ug0EfgSLR0f+K41eNqqWRftGR83emqPLDOf0ZLTLjVFGJaH
4Jn4l1gG+JDf/n/zMyahxtV7+mT8GOFFigFfuxJaxMjr2j7IvELlxQ4IfZgWwqCm4qQecv3
G+N9YdbjveMVXW0v4XwIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEAZyZAm8UmP9PQYwNy4P
WpeKAN1IZu4DOSLdM5VwXuJcShKxW7CyCVglUxizIF147HRW7kVnodq9r3BoWhbJ+FR2KZU
mypk79t2nvzKbwKVb97G+MT/gw1pLRsI1uBoKinMfLgKp1Q38K5Py2VGW1E47K7/rhm3yVQ
rIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

3. Copy the subject line from the temp file, and then paste it into the subject line of the new message.

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

5. Copy the certificate information from the temp file, including the headers `---BEGIN NEW CERTIFICATE REQUEST---` and `---END NEW CERTIFICATE REQUEST---`, and paste it into the body of the new message. For example:



6. Send the email message to the CA.

Once you have emailed your request, you must wait for the CA to respond with your certificate. Response time for your request is highly variable. For example, if your CA is internal to your company, it may only take a day or two to respond to your request. If your selected CA is external to your company, it could take several weeks to respond to your request.

When the CA sends a response, be sure to save the information in a text file. You will need the data when you install the certificate. If you are using client authentication with replication, you will also need to provide the Certificate Subject DN when you configure the servers for replication.

You should also back up the certificate data in a safe location. If your system ever loses the certificate data, you can reinstall the certificate using your backup file.

Once you receive your certificate, you are ready to install it in your server's certificate database as described in the next step.

Step 3: Install the Certificate

To install a server certificate:

1. On the Directory Server Console, select the Tasks tab.
2. Click Certificate Setup Wizard.

   A dialog box appears outlining the steps required to set up a server certificate. Click Next.
3. On the dialog box that appears, provide information as follows, and then click Next.

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

   Is the server certificate already requested and ready to install. Choose Yes.
4. A dialog appears confirming that the wizard is ready to continue with the certificate setup. Click Next.

5. The Install the Server Certificate - Step 1 dialog box appears. Provide the following information and then click Next.

   Certificate for. If you are installing your server's certificate choose "This Server." If you are installing your CA's certificate choose "Server Certificate Chain".

   You only choose "Trusted Certificate Authority" if you are using a certificate that you want to accept as a trusted CA for client authentication, as described in "Step 4: Trust the Certificate Authority" on page 306.

   Password. Enter the certificate database password you used when you generated the certificate request.

6. The Install the Server Certificate - Step 2 dialog box appears. Choose one of the following options and then click Next.

   The certificate is located in this file. You can either enter the absolute path to the certificate in this text box, or copy and paste the certificate as described below.

   The certificate is located in the following text field. Copy the text from the CAs email or from the text file you created and paste it in this field. For example:

```
-----BEGIN CERTIFICATE-----
MIICMjCCAZugAwIBAgICCEEwDQYJKoZIhvcNAQEFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoTGlBhbG9va2FWaWxsZSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTWFrZXJzICdSJyBVczEpMCcGA1UEAxMgVGVzdCBUZXN0IFRlc3QgVGVz
dCBUZXN0IFRlc3QgQ0EwHhcNOTgwMzEyMDIzMzU3WhcNOTgwMzI2MDIzMzU3WjBP
MQswCQYDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZWN0b3J5IFB1Ymxp
Y2F0aW9uczEWMBQGA1UEAxMNZHVgh49dq2itLmNvbTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYCQQCksMR/aLGdfp4m0OiGcgijG5KgOsyRNvwGYW7kfW+8mmijDtZRjYNj
jcgpF3VnlsbxbclX9LVjjNLC57u37XZdAgEDozYwNDARBglghkgBhvhCAQEEBAMC
APAwHwYDVR0jBBgwFoAU67URjwCaGqZuUpSpdLxlzweJKiMwDQYJKoZIhvcNAQEF
BQADgYEAJ+BVem3vBOP/BveNdLGfjlb9hucgmaMcQa98A/db8qimKT/ue9UGOJqL
bwbMKBBopsD56p2yV3PLJIsBgrcuSoBCuFFnxBnqSiTS/7YiYgCWqWaUAExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

Now that you have installed your certificate, you need to configure your server to trust the Certificate Authority from which you obtained the server's certificate. This process is described in the next step.

Step 4: Trust the Certificate Authority

This process consists of obtaining your CA's certificate and installing it into your server's certificate database. This process differs depending on the certificate authority you use. Some commercial CAs provide a link off of their web site that allows you to automatically download the certificate. Others will email it to you upon request.

Once you have received the CA certificate, use the Certificate Setup Wizard to configure the Directory Server to trust the Certificate Authority. To do this:

Use the Certificate Setup Wizard to configure the directory server to trust the Certificate Authority. To do this:

1. On the Directory Server Console, select the Tasks tab.
2. Click Certificate Setup Wizard.

   A dialog box appears outlining the steps required to set up a server certificate. Click Next.
3. In the screen that appears, select the same token you used when you generated the certificate request and then select Yes to indicate that you are ready to install a certificate. Click Next.
4. A summary screen appears indicating that the wizard is ready to proceed with the installation. Click Next.
5. The Install the Server Certificate - Step 1 dialog box appears. Select the Trusted Certificate Authority or Server Certificate Chain radio button as appropriate and then type the directory server's certificate database password. Click Next.
6. The Install the Server Certificate - Step 2 dialog box appears. If you saved the CA's certificate to a file, enter the path in the text box provided. If you received the CA's certificate via email, copy and paste the certificate including the headers into the text field provided. Click Next.
7. The Install the Server Certificate - Step 3 dialog box appears. Click Add to add the certificate to the trust database.
8. A confirmation dialog box appears. Click Done to exit the wizard.

Once you have installed your certificate and trusted the CA's certificate, you are ready to activate SSL; however, Netscape recommends that you confirm that the certificates have been installed correctly as described in the next step.

Step 5: Confirm That Your New Certificates Are Installed

1. On the Directory Server Console, select the Tasks tab.
2. Choose Manage Certificates from the Console menu.

   The Certificate Management dialog box that appears contains a list of all the installed certificates for the directory server.
3. Scroll through the list. You should find the certificates you installed. If you find the certificates, your server is ready for SSL activation.

Activating SSL

Most of the time, you want your server to run with SSL enabled. If you temporarily disable SSL, make sure you re-enable it before processing transactions that require confidentiality, authentication, or data integrity.

Before you can activate SSL, you must create a certificate database, obtain and install a server certificate and trust the CA's certificate as described in "Obtaining and Installing Server Certificates" on page 298.

To turn on SSL communications with your directory server:

1. Set the secure port you want the server to use for SSL communications. See "Changing Directory Server Port Numbers" on page 292 for information.

   The encrypted port number that you specify must not be the same port number you use for normal LDAP communications. By default, the standard port number is 389 and the secure port is 636.
2. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.

3. Select the Encryption tab in the right pane.

   This displays the current server encryption settings.

4. Indicate that you want encryption enabled by selecting the “Enable SSL” checkbox.

5. Select the checkbox next to the cipher family or families you want to use.

   You can specify which ciphers you want the server to use by selecting a cipher family and then clicking Cipher Preferences. For more information about specific ciphers, see “Setting Security Preferences” on page 309.

6. Select the token you want the server to use.

7. Select the certificate that you want to use.

   You create the encryption alias when you create your server’s certificate database. For more information about certificate databases, see the “Enabling SSL Encryption” section in *Managing Servers with Netscape Console*. For instructions on setting up a certificate database for your server, see “Obtaining and Installing Server Certificates” on page 298.

8. If you do not want the server to use client authentication, select “Do not allow client authentication”. If you want the server to use client authentication, select “Allow client authentication” or “Require client authentication” as appropriate.

   The default is “Allow client authentication”. For more information about certificate-based authentication, see ““Using Certificate-Based Authentication” on page 311”.

   If you are using certificate-based authentication with supplier-initiated replication, then you must configure the consumer server to either allow or require client authentication.

   If you are using certificate-based authentication with consumer-initiated replication, then you must configure the supplier server to either allow or require client authentication.

Warning Selecting "Require client authentication" will disable communication between the Netscape Console and the directory server. This is because the Netscape Console does not support client authentication. If you select this option, you will no longer be able to manage your Netscape Servers from the Netscape Console; instead, you must use the command-line tools.

9. If you want Netscape Console and the directory server to use SSL during communications, select Use SSL in Netscape Console.

10. Click Save.

11. Restart the Directory Server. See "Starting the Server with SSL Enabled" on page 30 for information.

Setting Security Preferences

You can choose the type of ciphers you want to use for SSL communications. A *cipher* is the algorithm used in encryption. Some ciphers are more secure or *stronger* than others. Generally speaking, the more bits a cipher uses during encryption, the more difficult it is to decrypt the key. (For a more complete discussion of algorithms and their strength, see *Managing Servers with Netscape Console*.)

When a client initiates an SSL connection with a server, the client tells the server what ciphers it prefers to use to encrypt information. In any two-way encryption process, both parties must use the same ciphers. There are a number of ciphers available. Your server needs to be able to use the ciphers that will be used by client applications connecting to the server.

You might not want to enable all ciphers in order to prevent SSL connections with less than optimal encryption. Under most circumstances, United States law prohibits the export of products with 128-bit encryption, so overseas clients might only be using 40-bit encryption, which is not as difficult to crack as 128-bit. Deselecting all 40-bit ciphers effectively restricts access to clients available only in the United States.

Domestic versions of the Directory Server provide the following SSL 3.0 ciphers:

- RC4 cipher with 128-bit encryption and MD5 message authentication.
- RC4 cipher with 40-bit encryption and MD5 message authentication.

- RC2 cipher with 40-bit encryption and MD5 message authentication.
- DES with 56-bit encryption and SHA message authentication.
- FIPS DES with 56-bit encryption and SHA message authentication. This cipher meets the FIPS 140-1 U.S. government standard for implementations of cryptographic modules.
- Triple DES with 168-bit encryption and SHA message authentication.
- FIPS Triple DES with 168-bit encryption and SHA message authentication. This cipher meets the FIPS 140-1 U.S. government standard for implementations of cryptographic modules.
- No encryption, only MD5 message authentication.

Export versions of the Directory Server provide the following SSL 3.0 ciphers:

- RC4 cipher with 40-bit encryption and MD5 message authentication.
- RC2 cipher with 40-bit encryption and MD5 message authentication.
- No encryption, only MD5 message authentication.

In addition, the directory server also provides FORTEZZA ciphers. For information on using FORTEZZA with the Directory Server, see Chapter 12, “Managing FORTEZZA.”.

To select the ciphers you want the server to use:

1. Make sure SSL is enabled for your server. For information on how to do this, see “Activating SSL” on page 307.
2. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
3. Select the Encryption tab in the right pane. This displays the current server encryption settings.
4. Select one or more cipher families you want to use and then click Cipher Preferences.

5. In the dialog that appears, specify which ciphers you want your server to use by selecting them in the list. Unless you have a security reason to not use a specific cipher, you should select all of the ciphers except for `NULL`. When you are finished, click OK.

6. On the Encryption tab, click Save.

Warning You might not want to select the none, MD5 cipher. If no other ciphers are available on the client, the server uses this, and no encryption occurs.

In order to continue using the Netscape Console with SSL, you must select at least one of the ciphers listed next.

For the export version of the server:

- RC4 cipher with 40-bit encryption and MD5 message authentication.
- No encryption, only MD5 message authentication.

For the domestic version of the server:

- Any of the ciphers required for the export version of the server.
- DES with 56-bit encryption and SHA message authentication.
- RC4 cipher with 128-bit encryption and MD5 message authentication.
- Triple DES with 168-bit encryption and SHA message authentication.

Using Certificate-Based Authentication

LDAP clients can bind to the directory server using certificates rather than normal Bind DN/Password authentication. This kind of authentication provides two things:

- Under some circumstances it is more convenient to provide a certificate for authentication purposes than to continually provide Bind DN/password credentials. This is true for situations where you are using applications that prompt you once for your certificate database password, and then use that certificate for all subsequent bind or authentication operations.

- The use of certificate-based authentication is more secure than non-certificate bind operations. This is because certificate-based authentication uses public-key cryptography. As a result, bind credentials cannot be intercepted across the network.

The directory server allows you to use certificate-based authentication using the command-line tools, for replication communications, and for applications you write using the LDAP SDK.

To set up certificate-based authentication, you must:

1. Create a certificate database for both the client and the server.

   In the case of supplier-server to consumer-server replication, you need a certificate database for both servers.

2. Obtain a certificate for both client and server.

   For replication communications, you obtain server certificates. For command-line tools, you obtain a client certificate.

3. Map the certificate's distinguished name to a distinguished name known by the directory server.

   This allows you to set access control for the client when it binds using this certificate. This mapping process is described in the "Mapping Client Certificates to LDAP" section in *Managing Servers with Netscape Console.*

For information on creating a certificate database and obtaining a certificate for use with replication, see See "Obtaining and Installing Server Certificates" on page 298. For command-line clients, see See "Creating Certificate Databases for LDAP Clients" on page 313.

Warning Requiring client authentication disables communication between the Netscape Console and the directory server. This is because the Netscape Console does not support client authentication. If you configure the server to require client authentication, you will no longer be able to manage your Netscape Servers from the Netscape Console; instead, you must use the command-line tools.

Creating Certificate Databases for LDAP Clients

If you want to use SSL and/or certificate-based authentication with LDAP clients such as ldapmodify, ldapsearch, or the NT synchronization service, you must:

- Create a certificate database.
- Trust the Certificate Authority (CA) that issues the server ceritificate.
- Optionally obtain and install a personal certificate (this is necessary only if you are going to use certificate-based client authentication instead of simple bind/password authentication).

The following procedure describes how to use Netscape Communicator 4.x to perform these activities.

Note You must use Netscape Communicator version 4.x to create a certificate database for clients that are communicating with Netscape Directory Server 4.x.

1. For Communicator under Unix operating systems, create a fresh user account and run Communicator from there. This creates a fresh certificate database to be created for that user account.

   Under Windows NT, you should create a new user profile for the purpose of obtaining the certificate database. You can create a user profile by using the Netscape User Profile Manager tool. It is available by default from the Start menu under `Programs -> Netscape Communicator -> Utilities`.

2. Use Communicator to connect to your Certificate Authority (CA). If you are using an internally deployed Netscape Certificate Server, you will go to a URL of the form:

   `https://<hostname>:444`

3. Trust the CA. This task differs depending on the CA. In some cases, such as if you are connecting to a Netscape Certificate Server, Communicator will automatically prompt you to see if you want to trust the CA. Other CA's provide a link that allows you to download the CA's certificate.

4. Optionally obtain a client certificate from the CA. This is required only if your client will use certificate-based authentication. Again, how you do this depends on your CA, but most will provide a link to a form that allows you to request the certificate. Usually, but not always, the certificate is mailed to you once it is assigned by the CA. This process can take from a few moments to several weeks.

   When you receive your certificate, install it in your certificate database file. Regardless of how you receive your certificate (either in email or on a web page), there should be a link that you click to install the certificate. Click it and step through the dialog boxes that Communicator presents to you.

   Make sure you keep a record of the information that is sent to you with your certificate. In particular, you must know what your certificate's subject DN is so that it can be mapped to an entry in the directory. If you lose your subject DN, you will have to request a certificate all over again.

5. When you have trusted your CA and you have installed your certificate (if any), shut down Communicator and move your `cert7.db` and `key3.db` files to a location that is convenient for use with your LDAP clients. If you move these files using FTP, be sure to use a binary transfer.

   Under Unix operating systems, these files are available in the `.netscape` directory in your home directory.

   Under Windows NT, you can find these files in the following location:

   `<communicator home>\users\<profile name>`

   where `<communicator home>` is the directory where you installed Netscape Communicator and `<user name>` is the profile that you used to obtain the certificate. If you are not using user profiles, then there will be no `<profile name>` directory. In this case, `cert7.db` and `key3.db` will be under the `users` directory.

6. If you are using certificate-based authentication, map the subject DN of the certificate that you obtained to the appropriate directory entry. This procedure is described in *Managing Servers with Netscape Console*.

You can now use SSL with your LDAP clients. For information on how to SSL with:

- ldapmodify see “SSL Parameters” on page 244
- ldapdelete see “SSL Parameters” on page 248
- ldapsearch see “SSL Parameters” on page 215
- the NT Synchronization Service, see the *Netscape Directory Server Installation Guide.*

Chapter

12

Managing FORTEZZA

The United States government developed FORTEZZA, an encryption system used by federal and government agencies to manage sensitive but unclassified information. Use the Server Console and the Certificate Setup Wizard to configure your server to work with FORTEZZA. (If you are not using FORTEZZA with the Directory Server, the options are not available in the Server Console.) For information on installing the FORTEZZA hardware, see the documentation that came with your card reader.

This chapter includes the following sections:

What You Need To Do

Before you can use FORTEZZA with the Directory Server, you need to complete the following:

1. Install the FORTEZZA PKCS #11 module provided with the Netscape Administration Server. See *Managing Servers with Netscape Console* for information.
2. Use the Certificate Setup Wizard to create a trust database. See "Setting Up FORTEZZA" on page 318 for more information.
3. Activate FORTEZZA as described in "Activating FORTEZZA" on page 320.
4. Restart the Directory Server as described in "Starting the Server with FORTEZZA Enabled" on page 322.
5. Optional. If you are using client authentication with the Directory Server, you also need to trust the CAs used by any clients that will be querying the Directory Server.
6. Optional. Enable SSL for the Administration Server as described in *Managing Servers with Netscape Console*.

Setting Up FORTEZZA

This section walks you through the process of configuring your server to work with FORTEZZA. This process is a necessary first step before you activate FORTEZZA in your directory. (For information on activating FORTEZZA in your Directory Server, see "Activating FORTEZZA" on page 320.) Before you begin, ensure that the FORTEZZA hardware is correctly connected to the Directory Server's host computer.

This walkthrough consists of the following steps:

- Step 1: Install the FORTEZZA PKCS #11 Module
- Step 2: Create a Trust Database

Step 1: Install the FORTEZZA PKCS #11 Module

Before you can use FORTEZZA with your Directory Server, you must first install the FORTEZZA PKCS #11 module. For more information, refer to the online guide *Managing Servers with Netscape Console* before continuing with this procedure.

Step 2: Create a Trust Database

A Trust Database is a key-pair and trust database installed on the local host. When you use an external token and the external device has insufficient storage capacity, the local Trust Database stores your Certificate Revocation Lists (CRLs), certificate chains, and trusted CA information.

If you have already set up a Trust Database for the server's host, skip to "Activating FORTEZZA" on page 320.

To create the trust database:

1. On the Directory Server Console, select the Tasks tab and click Certificate Setup Wizard. A dialog box appears outlining the steps required to set up a server certificate. Click Next.
2. On the dialog box that appears, choose the default from the "Select a token (Cryptographic Device)" pull-down menu.
3. Under "Is the server certificate already requested and ready to install", select "Do not install a certificate".

   With FORTEZZA, your key is stored in an external device. Although you do not need to install a certificate, you do need to run the New Trust Database Setup program once.
4. Click Next.
5. Click Next again to create the trust database.

6. Enter and confirm the password you want to use for the trust database and click Next.

   The password must contain at least 8 characters, at least one of them numeric. This password helps secure access to the new key database you are creating.

7. A dialog appears confirming that the trust database has been created. Click Done to dismiss the Certificate Setup Wizard.

Activating FORTEZZA

Most of the time, you want your server to run with FORTEZZA enabled. If you temporarily disable FORTEZZA, make sure you re-enable it before processing transactions that require confidentiality, authentication, or data integrity.

For more information about using FORTEZZA with the Directory Server, refer to Chapter 5 “Using SSL” of the online guide *Managing Servers with Netscape Console* and “Setting Up FORTEZZA” on page 318 in this manual before continuing with this procedure.

To turn on FORTEZZA communications in your Directory Server:

1. Make sure the hardware is attached correctly and put the card in the slot from which you want your server to read.

2. Create a trust database for the Directory Server. See “Setting Up FORTEZZA” on page 318 for more information.

3. Set the secure port you want the server to use for secure communications. See “Changing Directory Server Port Numbers” on page 292 for information.

   The encrypted port number that you specify must not be the same port number you use for normal LDAP communications.

4. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.

5. Select the Encryption tab in the right pane. This displays the current server encryption settings.

6. Indicate that you want encryption enabled by selecting the “Enable SSL” checkbox.

7. Select the checkbox next to the cipher family or families you want to use.

   In order to use FORTEZZA, you must select the FORTEZZA checkbox and then select one or more of the FORTEZZA ciphers.

8. Click Cipher Preferences.

   The Cipher Preferences dialog box displays. You must select at least one FORTEZZA cipher to activate FORTEZZA. Click OK to return to the Encryption tab when you are finished.

   The Directory Server provides the following SSL 3.0 FORTEZZA ciphers:

   - FORTEZZA with 80-bit Skipjack encryption and SHA message authentication. Skipjack is a data encryption and decryption algorithm. For added security, FORTEZZA ciphers use SHA message authentication. SHA is a government standardized algorithm used to construct a message authentication code that detects attempts to modify data while it is in transit. SHA is slower than MD5, but it is stronger.

   - FORTEZZA with 128-bit RC4 encryption and SHA message authentication. This cipher has approximately 10^{38} possible keys, making it very difficult to crack.

   - No encryption, only FORTEZZA/SHA message authentication. This cipher uses only SHA message authentication to secure data. Any data sent using this cipher is not encrypted. The data is protected from modification, but it can be viewed by eavesdroppers.

9. Select the token, or card slot, you want the server to use.

10. Enter the certificate, or personality, that you want to use in the “Certificate” text box. This certificate is stored on the FORTEZZA card.

11. If you want the server to use client authentication, select “Allow client authentication” or “Require client authentication” as appropriate. For more information about certificate-based authentication, see “Using Certificate-Based Authentication” on page 311.

12. Click Save.

13. Restart the Directory Server. See “Starting the Server with FORTEZZA Enabled” on page 322 for information.

Warning Requiring client authentication disables communication between Netscape Console and the directory server. This is because Netscape Console does not support client authentication. If you configure the server to require client authentication, you will no longer be able to manage your Netscape Servers from Netscape Console; instead, you must use the command-line tools.

If you want the Directory Server and Administration Server to use FORTEZZA for communications, you need to set up FORTEZZA for the Administration Server. See *Managing Servers with Netscape Console* for information.

Starting the Server with FORTEZZA Enabled

If you are using FORTEZZA on Windows NT, you can start the Directory Server from the Server Console. For all platforms, you can start the server from the command line. In either case, you must start the server from the physical machine where you installed the Directory Server.

This section explains:

- “Starting a FORTEZZA-Enabled Server From the Server Console (Windows NT Only)” on page 323
- “Starting a FORTEZZA-Enabled Server From the Command Line” on page 323

Starting a FORTEZZA-Enabled Server From the Server Console (Windows NT Only)

To start a FORTEZZA-enabled Directory Server from the Server Console on Windows NT:

1. On the server's host machine, make sure the hardware is attached correctly and put the card in the slot from which you want your server to read.

   You must start the Directory Server from the physical machine where you installed the Directory Server.

2. On the Directory Server Console, select the Tasks tab.

3. Click "Start the Directory Server".

4. When prompted, enter the PIN for the FORTEZZA card.

   The PIN number is packaged with the FORTEZZA crypto card and is not provided by Netscape. If you are also using other SSL cipher-families, you will also be prompted for the trust database (internal token) password.

Starting a FORTEZZA-Enabled Server From the Command Line

To start a FORTEZZA-enabled Directory Server from the command line:

1. On the server's host machine, make sure the hardware is attached correctly and put the card in the slot from which you want your server to read.

   You must start the Directory Server from the physical machine where you installed the Directory Server.

2. At the command prompt, type `<NSHOME>/slapd-<serverID>/start-slapd` and press Enter. Make sure you replace `<NSHOME>` with the server root directory and `<serverID>` with the name of the Directory Server.

3. When prompted, type the PIN number for the FORTEZZA card and press Enter.

   The PIN number is packaged with the FORTEZZA crypto card and is not provided by Netscape. If you are also using other SSL cipher-families, you will also be prompted for the trust database (internal token) password.

Disabling FORTEZZA

You disable FORTEZZA by configuring the server not to use the FORTEZZA cipher family for encrypted communications. Disabling FORTEZZA will disable SSL if you are using only FORTEZZA ciphers. For information on disabling SSL, see “Activating SSL” on page 307. To enable or disable FORTEZZA for your server:

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
2. Select the Encryption tab in the right pane. This displays the current server encryption settings.
3. To disable FORTEZZA, clear the FORTEZZA cipher family checkbox. For information on enabling FORTEZZA, see “Activating FORTEZZA” on page 320.
4. Click Save.
5. Restart the Directory Server. For information on how to do this, see “Starting the Server with FORTEZZA Enabled” on page 322 or “Starting and Stopping the Directory Server” on page 29 as appropriate.

Specifying FORTEZZA Options

You can configure the following options for the Directory Server:

- The name of the personality (certificate) used in key exchange
- The trust database password
- Compromised key list (CKL) and Certificate Revocation List (CRL) file locations.

For information on how to manage these options, see “Managing Server Certificates” in Chapter 5 “Using SSL” of *Managing Servers with Netscape Console.*

Using FORTEZZA With Client Authentication

To use FORTEZZA with client authentication:

- You must install a current Compromised Key List (CKL). For information, see “Managing Certificate Lists” in Chapter 5, “Using SSL” of *Managing Servers with Netscape Console*.
- You must install and configure a trusted CA for the FORTEZZA PKF hierarchy. See Appendix C, “FORTEZZA” of *Managing Servers with Netscape Console*.
- You also may need to configure the certificate mapping services of the Directory Server. See “Using Client Certificates” in Chapter 5, “Using SSL” of *Managing Servers with Netscape Console*.

Chapter

13

Managing Replication

Replication is an important mechanism for extending your directory service beyond a single server configuration. In the following sections, this chapter describes how you can use replication for your directory service:

For conceptual information on how you can use replication in your directory service, see the *Netscape Directory Server Deployment Guide*.

Replication Overview

Replication is the mechanism by which directory data is automatically copied from one directory server to another. Using replication, you can copy entire directory trees or subtrees between servers. Updates of any kind—entry additions, modifications, or even deletions—are automatically mirrored to other directory servers using replication.

The master copy of all directory data is stored in one and only one directory location. The server that controls this master copy of directory data is called the *supplier* server. Only the supplier server can modify or delete data in the master copy. The supplier server can then propagate changes made to its directory data to other servers known as *consumer* servers.

There are two basic forms of replication available to you: *supplier-initiated replication* and *consumer-initiated replication*. Supplier-initiated replication allows you to configure a supplier server to push data to one or more consumer servers. Consumer-initiated replication allows you to configure consumer servers to pull directory data from a supplier server.

Directory servers use replication agreements to define replication. A *replication agreement* identifies the directory objects to replicate, the times during which replication can occur, and the server to which the replicated data is pushed or the server from which replicated data is pulled.

This chapter provides detailed information on how to set up replication agreements. For more overview information on what replication is and how you might use it, see the *Netscape Directory Server Deployment Manual.*

Managing Supplier-Initiated Replication (SIR)

This section provides information on how you can manage SIR agreements in the following sections:

For more information about how to use replication within your enterprise, see the *Netscape Directory Server Deployment Guide*.

Configuring Servers for SIR

Before you can create an SIR agreement, you have to configure your servers to be either a supplier or a consumer. To do this, you must configure basic information about the servers.

To configure servers for supplier-inititated replication you need to do the following:

- On the consumer server—Configure a DN and a corresponding password for the supplier to use to bind to the consumer.

  If you want communication between the consumer and supplier to take place over SSL you also need to configure a certificate subject DN on the consumer server. For specific instructions, see “Configuring the Supplier DN for SIR” on page 329.

- On the supplier server—Configure a change log directory and then restart the supplier server. For specific instructions, see “Configuring the Change Log for SIR” on page 331.

  The change log is a special directory maintained by the supplier server that identifies the changes made to the server’s primary directory tree.

Once you have set up the consumer and supplier servers, you are ready to create the agreement. For specific instructions, see “Creating an SIR Agreement” on page 332.

Configuring the Supplier DN for SIR

Before you configure an SIR agreement, you need to configure a supplier DN and password on your consumer server. The supplier server uses this DN to bind to the consumer server during replication. The Supplier DN is a special distinguished name that does not actually exist in your directory tree. Instead, it is identified by the Supplier DN parameter in the `slapd.conf` file.

Entries supplied to the consumer server from another server can only be updated if the LDAP client binds as the supplier DN; all other update operations for the supplied data will be referred to the supplier server that masters the directory data. It is therefore important that the only LDAP clients that bind to this server using the supplier DN are supplier servers.

You can configure the consumer server to accept simple or password-based bind operations from the supplier, and you can configure the server to accept certificate-based authentication. Certificate-based authentication is more secure because the password used to authenticate is encrypted and does not need to be stored in cleartext on the supplier server. To configure the supplier DN and authentication method:

1. On the consumer server's Directory Server Console, select the Configuration tab and then select the Replication Agreements folder.
2. Select the Consumer Server Settings tab in the right pane.
3. In order to use simple authentication or certificate-based authentication, you must enter the DN and password you want the supplier server to use to bind to this server in the Supplier DN, and Supplier password text boxes.
4. If you want to use certificate-based authentication, type or paste the subject DN of the certificate that the supplier server will use to bind to this server in the Supplier Certificate Subject DN box.

   If you have more than one server supplying entries to this consumer, enter a subject DN for each supplier server. Each DN should be placed on a separate line in the box.

   You can find the subject DN of the certificate used by a supplier server from the supplier server's console. For specific instructions on how to do this, see "Setting up Encryption Security" in *Managing Servers with Netscape Console*.

Configuring the Change Log for SIR

Before a server can supply directory entries to consumer servers, you must configure a change log on the supplier server. The change log is a special database maintained by the supplier server that identifies the changes made to the server's primary directory tree. To configure the change log:

1. On the supplier server's Directory Server Console, select the Configuration tab and then select the Replication Agreements folder.
2. Select the Supplier Server Settings tab in the right pane.
3. Type the full path to the directory where you want the server to store the change log in the Changelog Database Directory text box.

   This directory must be located on the supplier's local disk. If you want the directory server to suggest a pathname, click Use Default.
4. Enter a DN to use as the change log's directory suffix in the Changelog Suffix text box. Typically, this suffix is: `cn=changelog`.
5. Either enter the maximum number of records you want the change log to record in the Max Changelog Records text box, or if you do not want to set a maximum number of entries for the change log, select Unlimited.
6. If you want the server to remove entries from the change log after they reach a certain age, specify that age in seconds, minutes, hours, days, or weeks in the Max Changelog Age fields.

   If you do not want to configure a maximum age, select Unlimited; the server will not remove entries from the change log due to their age.
7. Click Save.
8. Restart the directory server.

You are now ready to configure an SIR agreement.

Creating an SIR Agreement

To create a supplier-initiated replication agreement:

1. On the Directory Server Console, select the Configuration tab.
2. Right-click the Replication Agreements folder and select New Replication Agreement.

   The Replication Agreement Wizard appears. This wizard takes you through the steps of setting up a replication agreement.
3. On the wizard dialog, select Supplier Initiated Agreement and click Next.

   A dialog box appears that allows you to provide a name for the replication agreement.
4. Provide a name for the replication agreement and click Next.

   The Replication Agreement form displays.
5. Select a consumer from the Consumer drop-down menu or, click Other to manually enter the host and port number of the consumer server.
6. If you want the servers to use SSL during replication, select the "Using encrypted SSL connection" checkbox.
7. If you want the servers to use SSL client authentication, select SSL Client Authentication.

   You cannot use SSL Client Authentication unless you have specified that the server use encrypted SSL connections in Step 6.
8. If you want the servers to communicate using simple authentication (with or without SSL), select Simple Authentication and then provide the Bind DN and password.
9. Enter the subtree you want replicated in the Subtree text box or, choose Browse to select the node you want to replicate.
10. When you are finished, click Next. This brings up the Replication Schedule form.

11. If you do not want to limit replication to explicit time periods, select "Always keep directories in sync". Alternatively, you can identify the time of day and day(s) of the week when replication can occur by selecting "Sync on the following days".

If you choose to limit replication to specific days and times, select the checkbox next to the day(s) and enter the hours between which replication can take place. Any replication activity occurring when the specified time interval ends will be completed, but no new replication processes will be started outside the specified replication interval.

When you are finished scheduling the replication agreement, click Next. The Consumer Initialization dialog displays.

12. Choose one of the following options:

- Do Not Initialize Consumer—Select this option if you do not want the consumer initialized automatically or the LDIF file created.
- Initialize Consumer Now—Select this if you want the server to initialize the consumer when you finish creating the replication agreement. For performance reasons, this is not recommended for databases larger than 10,000 entries.
- Create Consumer Initialization File—Select this if you want the server to export the replicated tree to LDIF so you can manually import it to the consumer. If you choose to have the server export to LDIF, supply the LDIF filename in the field provided.

13. Click Next.

The summary dialog appears.

You need to initialize the consumer before replication can occur. If you choose not to initialize the consumer now, see "Initializing Consumers" on page 344 for instructions on how to do it later.

14. Make sure that the information on the summary dialog box is correct.

If any information is incorrect, click Back to step back through the dialogs and change the information. When you are finished, click Done. The server creates the replication agreement and dismisses the replication wizard.

Duplicating an SIR Agreement

To add a consumer to a supplier-initiated agreement you need to duplicate the SIR Agreement and then update the consumer setting to point to the new consumer. In reality, you are not adding a consumer to an existing agreement, so much as copying the details of an existing agreement and adding the new consumer to the copy. To do this:

1. On the Directory Server Console, select the Configuration tab.
2. Open the Replication Agreements folder and then the Supplier Initiated Agreements folder.
3. Right-click the replication agreement in the tree and select Duplicate Replication Agreement from the pop-up menu.

   The Replication Agreement Wizard appears with the original settings displayed in the dialog boxes.
4. Enter a unique name representative of this agreement and click Next.
5. On the dialog that appears, enter the new consumer in the Consumer text box.
6. Complete the forms and click Done when you are finished.

Editing an SIR Agreement

You can make changes to existing replication agreements using the Directory Server Console. To do this:

1. On the supplier server's Directory Server Console, select the Configuration tab.
2. Open the Replication Agreements folder and then the Supplier Initiated Agreements folder.
3. Select the replication agreement you want to edit.
4. To change the name of the replication agreement, select the Summary tab in the right pane and enter your changes in the Agreement Name text box.

5. To edit the scheduling information for the replication agreement, select the Schedule tab in the right pane.

   If you do not want to limit replication to explicit time periods, select "Always keep directories in sync". Alternatively, you can identify the time of day and day(s) of the week when replication can occur by selecting "Sync on the following days". If you choose to limit replication to specific days and times, select the checkbox next to the day(s) and enter the hours between which replication can take place. Any replication activity occurring when the specified time interval ends will be completed, but no new replication processes will be started outside the specified replication interval.

6. To change the general settings for this replication agreement, such as the consumer server you want to replicate to, what you want replicated, and whether or not SSL is used for the connection, select the Content tab in the right pane.

   - To change the consumer this supplier replicates to, select a different consumer from the Consumer drop-down menu or, click Other to manually enter the host and port number of the new consumer server.
   - If you want the servers to use SSL during replication, select the "Using encrypted SSL connection" checkbox.
   - If you want the servers to use SSL client authentication, select SSL Client Authentication. You cannot use SSL Client Authentication unless you have specified that the server use encrypted SSL connections. In order for you to select this option, you must first configure SSL for both your supplier and consumer servers (see Chapter 11, "Managing SSL"), and configure your consumer server to recognize the subject DN your supplier server's certificate as the supplier DN (see "Configuring the Supplier DN for SIR" on page 329).
   - If you want the servers to communicate using simple authentication (with or without SSL), select Simple Authentication and then provide the Bind DN and password.
   - You can change the subtree you want replicated in the Content Replicate text box. Alternatively, choose Browse to browse the contents of the supplier server. If you are going to replicate a subtree, you must

make sure the appropriate parent entry is available on the consumer server. For example, if you are replicating the `ou=people, o=airius.com` subtree, then you must first make sure the consumer server contains the `o=airius.com` entry.

7. When you have finished making changes, click Save.

Managing Consumer-Initiated Replication (CIR)

This section provides information on managing CIR agreements in the following sections:

- "Configuring Servers for CIR" on page 336
- "Creating a CIR Agreement" on page 339
- "Editing a CIR Agreement" on page 341

For more information about how to use replication within your enterprise, see the *Netscape Directory Server Deployment Guide*.

Configuring Servers for CIR

Before you can create a CIR agreement, you have to configure your servers to be either a supplier or a consumer. To do this, you must configure basic information about the servers.

To configure servers for consumer-inititated replication you need to:

- Configure a change log directory on the supplier server and then restart the supplier server. For specific instructions, see "Configuring the Change Log for CIR" on page 337.
- Set up consumer access to the change log directory on the supplier server. This includes creating a DN for the consumer to use to connect to the supplier server and setting the access control instructions for the DN to

allow the consumer to search and read the change log. For specific instructions, see “Providing Consumer Access to the Change Log for CIR” on page 338.

Once you have set up the consumer and supplier servers, you are ready to create the agreement as described in “Creating a CIR Agreement” on page 339.

Configuring the Change Log for CIR

Before the consumer server can collect updated information from the supplier server, you must configure a change log for the supplier server. The change log is a special database maintained by the supplier server that identifies the changes made to the server's primary directory tree.

To configure the change log:

1. On the supplier server's Directory Server Console, select the Configuration tab and then select the Replication Agreements folder.
2. Select the Supplier Server Settings tab in the right pane.
3. In the Changelog Database Directory text box, type the full path to the directory where you want the server to store the change log.

   This directory must be located on the supplier's local disk. If you want the directory server to suggest a pathname, click Use Default.
4. In the Changelog Suffix text box, enter a DN to be used as the change log's directory suffix. Typically, this suffix is: `cn=changelog`.
5. Either enter the maximum number of records you want the change log to record in the Max Changelog Records text box, or if you do not want to set a maximum number of entries for the change log, select Unlimited.
6. If you want the server to remove entries from the change log after they reach a certain age, specify that age in seconds, minutes, hours, days, or weeks in the Max Changelog Age fields. If you do not want to configure a maximum age, select Unlimited; the server will not remove entries from the change log due to their age.
7. Click Save.
8. Restart the directory server.

Providing Consumer Access to the Change Log for CIR

Before you can use a consumer-initiated replication agreement, the consumer server must be able to read the change log directory.

To provide consumer access to the change log for CIR:

1. Create a directory entry (pseudo-user) that can be used by your consumer servers to read your change log.

   This directory entry does not have to be created in your change log directory tree; it can be a normal entry in your primary directory tree provided the entry contains the userPassword attribute. For information, see Chapter 9, “Managing Directory Entries.”

2. At the root level of your change log tree, create an ACI statement that grants the user from step 1 read, search, and compare access to the entire change log tree. In addition, this ACI should grant full read, search, and compare privileges for the tree or subtree that the consumer server will retrieve from the supplier server. For more information, see “Configuring the Change Log for CIR” on page 337 and Chapter 5, “Managing Access Control.”

For security reasons, Netscape recommends that you do not configure anonymous access for your change log directory tree. Also, you should grant only read, search, and compare access to the DN with which your consumers bind to your supplier; do not provide any form of write or delete access to this tree.

Finally, for logging and auditing purposes, you may want to configure a different directory entry (pseudo-user) for each consumer server. This allows you to track which consumer is binding to your server and when. For information on tracking access, see “Viewing the Access Log” on page 264.

Creating a CIR Agreement

To create a consumer-initiated replication agreement:

1. On the consumer server's Directory Server Console, select the Configuration tab.
2. Right-click the Replication Agreements folder and select New Replication Agreement.

   This brings up the Replication Agreement Wizard which takes you through the steps of setting up a replication agreement.
3. On the wizard dialog box, select Consumer Initiated Agreement and click Next.

   This displays a dialog that allows you to provide a name for the replication agreement.
4. Enter a name for the replication agreement and click Next.

   This displays the Replication Agreement dialog box.
5. Select the supplier server from which you want the consumer to pull replicated information or, click Other to manually enter the host and port number of the supplier server.
6. If you want the servers to use SSL during replication, select the "Using encrypted SSL connection" checkbox.
7. If you want the servers to use SSL client authentication, select SSL Client Authentication. You cannot use SSL Client Authentication unless you have specified that the server use encrypted SSL connections in Step 6.
8. If you want the servers to communicate using simple authentication (with or without SSL), select Simple Authentication and then provide the Bind DN and password.
9. Enter the subtree you want to replicate in the Content Replicate text box. Make sure the parent of the subtree exists on the consumer server. You can also click Browse to browse the contents of the supplier server. When you are finished, click Next. The Replication Schedule dialog box appears.

10. You must configure the consumer server to periodically check the supplier server to see if there are any pending updates by entering a time interval in the Update Interval text box. The interval is defined in minutes.

11. If you do not want to limit replication to explicit time periods, select "Always keep directories in sync". Alternatively, you can identify the time of day and day of week when replication can occur by selecting "Sync on the following days". If you choose to limit replication to specific days and times, select the checkbox next to the day(s) and enter the hours between which replication can take place. Any replication activity occurring when the specified time interval ends will be completed, but no new replication processes will be started outside the specified replication interval. When you are finished scheduling the replication agreement, click Next.

12. Choose one of the following options:

    - Do Not Initialize Consumer—Select this option if you do not want the consumer initialized automatically or an LDIF file created.

    - Initialize Consumer Now—Select this if you want the server to initialize the consumer when you finish creating the replication agreement. This is not recommended for databases larger than 10,000 entries.

13. Click Next. The summary dialog appears.

    You need to initialize the consumer before replication can occur. If you choose not to initialize the consumer now, see "Initializing Consumers" on page 344 for instructions on how to do it later.

14. Make sure that the information in the dialog box is correct. If any information is incorrect, click Back to step back through the dialogs and change the information. When you are finished, click Done. The server creates the replication agreement and dismisses the replication wizard.

Duplicating a CIR Agreement

To add a supplier to a consumer-initiated agreement you need to duplicate the CIR Agreement and then update the supplier setting to point to the new supplier. In reality, you are not adding a supplier to an existing agreement, so much as copying the details of an existing agreement and adding the new supplier to the copy. To do this:

1. On the Directory Server Console, select the Configuration tab.
2. Open the Replication Agreements folder and then the Consumer Initiated Agreements folder.
3. Right-click the replication agreement in the tree and select Duplicate Replication Agreement from the pop-up menu. This brings up the Replication Agreement Wizard with the original settings displayed in the dialog boxes.
4. Enter a unique name representative of this agreement and click Next.
5. On the dialog that appears, enter the new supplier in the Supplier text box.
6. Complete the rest of forms and click Done when you are finished.

Editing a CIR Agreement

You can make changes to existing replication agreements using the Directory Server Console. To do this:

1. On the consumer server's Directory Server Console, select the Configuration tab.
2. Open the Replication Agreements folder and then the Consumer Initiated Agreements folder.
3. Select the replication agreement you want to edit.
4. To change the name of the replication agreement, select the Summary tab in the right pane and enter your changes in the Agreement Name text box.

5. To edit the scheduling information for the replication agreement, select the Schedule tab in the right pane.

    1. You must configure the consumer server to check the supplier server to see if there are any pending updates. To do this, enter a time interval in the Update Interval text box.

    2. If you do not want to limit replication to explicit time periods, select "Always keep directories in sync". Alternatively, you can identify the time of day and day of week when replication can occur by selecting "Sync on the following days". If you choose to limit replication to specific days and times, select the checkbox next to the day(s) and enter the hours between which replication can take place. Any replication activity occurring when the specified time interval ends will be completed, but no new replication processes will be started outside the specified replication interval.

6. To change the general settings for this replication agreement, such as the supplier server you want to replicate from, what you want replicated, and whether or not SSL is used for the connection, select the Content tab in the right pane.

    1. To change the supplier from which this consumer gets information, select a different supplier from the Supplier drop-down menu. Alternatively, click Other to manually enter the host and port number of the new supplier server.

    2. If you want the servers to use SSL during replication, select the "Using encrypted SSL connection" checkbox.

    3. If you want the servers to use SSL client authentication, select SSL Client Authentication. You cannot use SSL Client Authentication unless you have specified that the server use encrypted SSL connections.

        In order for you to select this option, you must first configure SSL for both your supplier and consumer servers (see Chapter 11, "Managing SSL"), and configure your consumer server to recognize the subject DN or your supplier server's certificate as the supplier DN (see "Configuring the Supplier DN for SIR" on page 329).

4. If you want the servers to communicate using simple authentication (with or without SSL), select Simple Authentication and then provide the Bind DN and password.

5. You can change the subtree you want replicated in the Content Replicate text box. Alternatively, choose Browse to browse the contents of the supplier server.

7. When you have finished making changes, click Save.

Removing the Change Log

If you change the change log suffix, the entries in the change log are no longer valid. For this reason, you need to remove the change log and create a new one. If you remove the change log, you will need to reinitialize your consumer servers. You can remove the change log using the Directory Server Console. To do this:

1. On the supplier server's Directory Server Console, select the Configuration tab.

2. Select the Replication Agreements folder in the navigation tree in the left pane and then the Supplier Server Settings tab in the right pane.

3. Click Remove Changelog.

4. Click Save.

5. Restart the directory server.

6. Reinitialize your consumer servers.

Initializing Consumers

There are two ways that you can initialize a consumer. The options are described inthe following sections:

- “Online Consumer Creation” on page 346—This method is the easiest to perform but is prohibitively time consuming for databases that are larger than 5,000 - 10,000 entries in size.
- “Manual Consumer Creation” on page 348—This is the more difficult but most efficient method.

This section first describes consumer initialization in detail and then provides instructions on both consumer creation methods.

When to Initialize a Consumer

After you have created a replication agreement, you must initialize the consumer. That is, you must physically copy directory data from the supplier server to the consumer server so that future changes can be replayed to consumer servers.

Consumer initialization involves copying the replicated directory entries from the supplier server to the consumer server. When these entries are placed on the consumer server, the appropriate `copiedFrom` attribute must also be placed on the replicated tree (see “Replication Algorithms” on page 352 for details).

Once the tree has been physically placed on the consumer, the supplier server can begin replaying update operations to the consumer server (SIR) or the consumer can begin requesting data from the supplier (CIR).

In addition, any attempts to modify data on the consumer that is owned by the supplier are referred to the supplier server. For more information about referrals, see Chapter 14, “Managing Referrals.”

Under normal operations, the consumer should not ever have to be initialized again. However, there are several major events that can require a reinitialization of the consumer server:

- The supplier's database version number does not match the version number stored on the consumer for the replicated entries. This will happen if the supplier's database is either reloaded from a backup.

  The database version number is a unique identifier that allows the supplier and the consumer to know that the database has not been reloaded since the last synchronization.

- The change log on the supplier server is damaged to the extent that the supplier cannot determine what changes to replay to the consumer. This can happen if the supplier's change log becomes corrupted (such as might happen in the event of a disk failure) or if the change log is trimmed before the trimmed changes can be replayed to the consumer server.

  This situation is most likely to arise if the consumer's database is restored from a backup and the supplier's change log was truncated sometime after that backup was taken.

  Change logs are trimmed based on the Maximum Changelog Age and Maximum Changelog Size parameters. For more information, see "Configuring the Change Log for CIR" on page 337.

- The supplier server is continually repairing inconsistencies on the consumer server, or the supplier is unable to repair data inconsistencies on the consumer server.

  In almost all cases, once you have initialized the consumer server, the supplier can successfully repair inconsistencies on a consumer server. Further, even if it cannot repair data inconsistencies, the supplier will continue to replicate to the consumer server. When the supplier server detects a data inconsistency on a consumer server, the supplier issues the following message to the error log:

  ```
  Inconsistency detected while replaying change <n>,
  entry <DN>, to replica <host>:<port>/<DN>
  ```

This message also indicates whether the inconsistency could be repaired.

The process that you use to initialize or reinitialize a consumer differs depending on the type of consumer creation you use. See ""Online Consumer Creation" on page 346" or "Manual Consumer Creation" on page 348 for more information.

Note When a consumer server is being initialized via online consumer initialization, all operations (including searches) on the supplied tree are referred to the supplier server until the initialization process is completed.

Online Consumer Creation

Online consumer creation is the easiest way to initialize or reinitialize a consumer. However, this process can be very time consuming, and for large databases you may find that manual consumer creation is a more appropriate approach (refer to “Manual Consumer Creation” on page 348 for more information).

Online consumer creation works by moving data from the supplier to the consumer server over LDAP. That is, the replicated information is placed on the consumer server using LDAP add operations.

Before using online consumer creation, consider the performance implications of this method of consumer initialization. On a reasonably fast single processor (such as an Intel Pentium II or a Sun Sparc Ultra 1), you can expect online consumer creation to proceed at the following rates:

- For fresh initializations (that is, if the consumer server’s tree is empty), the supplier can add 9,000 to 36,000 entries per hour. The actual rate will depend on characteristics of your server such as the size of your entries, the amount of indexing your consumer is performing, the speed of your disk, the amount of RAM available to the consumer server, and the speed of your networks.
- For reinitializations, the supplier can add from 4,500 to 18,000 entries per hour. The performance drops by half because the online consumer creation process deletes all previously replicated entries from the consumer server before the consumer is initialized with a fresh set of data.

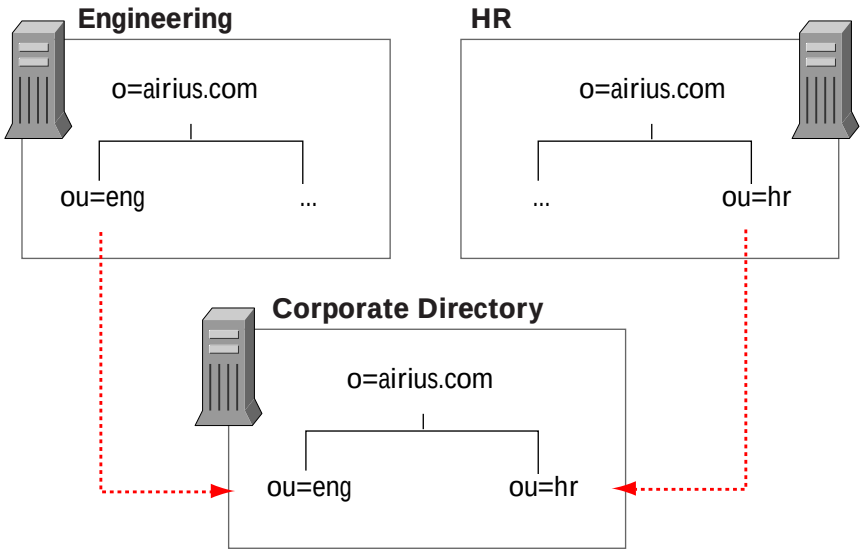
When You Should Use Online Consumer Creation

Essentially, you should always use online consumer creation unless you find the time that it takes to complete this operation objectionable.

You should always use online consumer creation when the consumer server has multiple subtrees supplied by different supplier servers, as shown in the figure below. In this case, you cannot use manual consumer initialization to

initialize the consumer because the import process used in manual consumer initialization replaces the entire database, including the subtrees supplied by other servers.

In the following figure, both Engineering and HR are supplier servers that replicate data to the main corporate directory.



How to Use Online Consumer Creation

To use online consumer creation:

1. Create a replication agreement. For details on creating replication agreements, see “Creating an SIR Agreement” on page 332 or “Creating a CIR Agreement” on page 339.
2. On the supplier server’s Directory Server Console, right-click the appropriate replication agreement on the Configuration tab and select Initialize Consumer from the pop-up menu.
3. Click Yes in the confirmation box.

Online consumer creation begins immediately. You can check the status of the online consumer creation on the console's Status tab. For more information about monitoring replication and initialization status, see "Monitoring Replication Status" on page 350. If online consumer creation is in progress, the status shows that a replica is being initialized. To update this window, click Refresh. When online consumer creation finishes, the status changes to reflect this.

You can configure your server to automatically reinitialize a consumer when the server detects an unexplainable inconsistency. To do this, place the following line in the `slapd.conf` file of either the supplier server (for supplier-initiated replication) or the consumer server (for consumer-initiated replication):

```
orcauto on
```

See "Directory Server Configuration Files" on page 36 for information on where the server stores `slapd.conf`.

The "Enable Online Consumer Creation" `slapd.conf` parameter causes the consumer to be reinitialized if a version number mismatch occurs between the supplier server's database and the replicated entries, or if the supplier is unable to replay changes to the consumer due to problems with the change log (see "When to Initialize a Consumer" on page 344 for more information).

Manual Consumer Creation

Manual consumer creation is the fastest method of consumer initialization for sites that are replicating very large numbers of entries. However, the manual process is more complicated than the online creation process.

You should use the manual process whenever you find that the online process is inappropriate due to performance concerns. However, you should never use the manual process if your consumer server contains directory data that is mastered by more than one directory server. That is, use the manual process only if your entire consumer server's database is supplied by a single supplier server. This is because typically when you use `ldif2db`, the database is completely overwritten. (The exception to this is the configuration tree `o=NetscapeRoot` which you can choose not to overwrite during initialization.)

For information on consumer initialization, see “Initializing Consumers” on page 344. For information on the online consumer creation process, see “Online Consumer Creation” on page 346.

To manually initialize or reinitialize a server:

1. Create a replication agreement as described in “Creating an SIR Agreement” on page 332 or “Creating a CIR Agreement” on page 339. When prompted, select the “Create consumer initialization file” radio button.

2. Import the LDIF file to the consumer server. See “Importing the LDIF File to the Consumer Server” on page 349 for instructions.

Converting the Supplier Tree to LDIF

You can convert the tree to LDIF when you create a replication agreement by selecting “Create consumer initialization file” on the Consumer Initialization dialog in the Replication wizard. (See “Creating an SIR Agreement” on page 332 or “Creating a CIR Agreement” on page 339 for information.)

If you choose not to export the tree at that time, you can:

- right-click the replication agreement in the Directory Server Console and select “Create LDIF File” from the pop-up menu, or

- use the export command as described in “Exporting Databases to LDIF” on page 70. If you convert the database to LDIF using the command-line, you must specify the `-r` argument to ensure that the `copiedFrom` attribute is included in the output.

Importing the LDIF File to the Consumer Server

Create your consumer server's database from the LDIF file by using either the Import command from the server console, or the `ns-slapd ldif2ldbm` command-line utility. For more information, see “Importing LDIF From the Server Console” on page 75, or “Importing LDIF From the Command Line” on page 77.

If your consumer server contains data that is also mastered either by itself or by some other supplier server, then use the online consumer creation process (for details, see “Online Consumer Creation” on page 346). While it is possible to

manually import this LDIF file, you must do so using `ldapmodify` which offers no performance improvement over online consumer creation because both mechanisms add entries over LDAP.

If you decide that you must manually initialize a consumer that contains data mastered by some other server than your supplier server, make sure you do the following:

- Create any entries that are parents of the replicated subtree on the consumer server before adding the replicated data. That is, if you are replicating `l=Minneapolis, ou=Global, o=airius.com`, make sure that you have created `ou=Global, o=airius.com` and `o=airius.com` on your consumer server before running the `add` operation.
- If you are reinitializing a consumer server, make sure you delete all of the contents of the replicated tree before running the `add` operation. If you do not delete the currently existing replicated tree, the server will fail the add operations, stating that the entries already exist.
- When you are adding or deleting replicated entries, bind to your consumer server using the supplier DN configured for that server. If you use any other DN (including the root DN), the consumer server will simply refer the modify operation to the supplier server.

Monitoring Replication Status

You can monitor replication status using the Directory Server Console.

To view a summary of replication status:

1. On the Directory Server Console, select the Status tab and then select Replication Agreements in the navigation tree in the left pane.

   In the right pane, a table appears that contains information about each of the replication agreements configured for this server.

2. Click Refresh to update the contents of the tab.

The status information displayed is described in Table 13.1.

Table 13.1 Directory Server Console - Status tab

Table Header	Description
Agreement	Contains the name you provided when you set up the replication agreement. A red bullet to the left of the name indicates an error has occurred and replication cannot take place. A green bullet indicates that replication is occurring normally. A yellow bullet indicates that all of the changes have not yet been sent to the consumer; this does not always indicate an error condition.
Supplier	Specifies the supplier server in the agreement.
Consumer	Specifies the consumer server in the agreement.
Change-Number	Indicates the last change number replayed to the consumer and the last change number available in the supplier's change log. For example: `[7] - [10]` "Unknown" indicates that the server has encountered an error and replication cannot continue or the server could not read one of the following: • The last change number from the supplier • The `copiedFrom` on the consumer These situations are normal if no changes have occurred on the supplier or if the consumer has not been initialized.
Status	Specifies the current state of the agreement. The possible values include: • Idle—No replication is currently taking place through this agreement. This might appear if there are no changes to replay or if the replication agreement is not scheduled to start until a later time. • Synchronizing—Changes are currently being sent to the consumer. • Populating—Online Replica Creation is in progress; the consumer is being initialized. • Halted—the synchronization process has encountered an error and quit.

Replication Algorithms

This section describes the replication processes in detail for both supplier-initiated and consumer-initiated replication.

SIR Algorithm

If you are using supplier-initiated replication, it is the responsibility of the supplier server to determine when its consumer servers need to be updated. This process occurs as follows:

1. Based on a schedule that you set, the supplier server determines that it is time to synchronize a consumer. The directory server identifies those subtrees that are replicated and the servers to which it is supplying those trees by using directory entries contained in the Machine data tree. Each consumer server is identified by a separate replication agreement, stored beneath the machine data entry. Part of each replication agreement is an identification of the root point of the replicated tree and a schedule indicating when the consumer should be updated.

2. If there are changes to replay, the supplier server binds to the consumer server using the supplier DN and password that you provide. The supplier must use this special DN for the bind or all updates to the replicated tree will be referred back to the supplier server.

   You use the Replication Agreements tabs to configure the supplier DN for a consumer server. For information, see “Configuring the Supplier DN for SIR” on page 329.

3. Each replicated tree contained on a consumer server includes a `copiedFrom` attribute that identifies the supplier of the subtree. This attribute is maintained by the replication subsystem. The supplier server examines the `copiedFrom` attribute in the replicated subtree's root point to ensure that no other supplier is identified by this attribute and to determine if replication can occur.

If no such attribute exists for the subtree, the supplier server

1. Writes the attribute to the replicated root point along with the appropriate attribute value
2. Aborts and immediately retries the synchronization

The syntax for the `copiedFrom` attribute is as follows:

```
copiedFrom: host:port generationID last_change
```

For example:

```
copiedFrom: dir.airius.com:389 019980610154028 12
```

where `host:port` is the host name and port number of the supplier server, `generationID` is a timestamp generated for the supplier database when the database is created, and `last_change` is a number generated by the supplier server that increases by one for every change replayed to the consumer. The `generationID` and `last_change` number are only reset if the database is reloaded. Do not modify the `generationID` or the `last_change` number manually. If you do, replication will fail and you will have to reinitialize your consumer servers.

Before replication can occur, the supplier checks the consumer's `copiedFrom` attribute value to ensure that:

- The host and port number match the current supplier.
- The database `generationID` matches the current supplier.
- The `last_change` number is smaller than the last change recorded in the supplier's change log. The change log is a log of all the changes made to the supplier's entries. Among other things, this log contains version identification used for synchronization purposes. For more information about configuring the change log for SIR, see "Configuring the Change Log for SIR" on page 331.

If the host and port do not match the current supplier, the supplier aborts synchronization and returns an error. This prevents any one replicated entry on the consumer from having multiple suppliers.

4. If no changes are required, the supplier terminates synchronization normally. If changes are required, the supplier updates and/or deletes all appropriate entries on the consumer as indicated by the change log. The supplier also records the last update number applied to the consumer and sets the `copiedFrom` attribute at the top of the replicated tree on the consumer server. This identifies the tree as being a replica and, more importantly, identifies the supplier server as the master of the information in that tree. The supplier then exits synchronization normally.

CIR Algorithm

If you are using consumer-initiated replication, it is the responsibility of the consumer server to request updates from the supplier server. This process occurs as follows:

1. Based on a schedule you set (which is stored in the consumer server's machine data tree), the consumer server determines that it wants to be updated and binds to the supplier server.

2. The consumer server obtains from its own replication agreements the location of each of its own directory trees supplied to it from another server. The consumer then examines the root point of each of these trees to make sure that a `copiedFrom` attribute is set on the root point.

   If no `copiedFrom` attribute exists on the tree, the consumer server adds one with the correct information so that all write operations are appropriately referred to the supplier server. This attribute is maintained by the replication subsystem.

   The syntax for the `copiedFrom` attribute is as follows:

   ```
   copiedFrom: host:port generationID last_change
   ```

   For example:

   ```
   copiedFrom: dir.airius.com:389 019980610154028 12
   ```

   where `host:port` is the host name and port number of the supplier server, `generationID` is a timestamp generated for the supplier database when the database is created, and `last_change` is a number generated by the supplier server that increases by one for every change replayed to the

consumer. The `generationID` and `last_change` number are only reset if the supplier database is reloaded. Do not modify the `generationID` or the `last_change` number manually. If you do, replication will fail and you will have to reinitialize your consumer servers.

3. The consumer server searches the supplier's change log directory, and compares the contents to its own `copiedFrom` attribute value to ensure:

   - The host and port number match the current supplier.
   - The database `generationID` matches the current supplier's generation ID number.
   - The `last_change` number is smaller than the last change recorded in the supplier's change log. The change log is a log of all the changes made to the supplier's entries. This log contains information used for synchronization purposes. For more information about configuring the change log for CIR, see "Configuring the Change Log for CIR" on page 337.

   If the host and port do not match the current supplier, the consumer aborts synchronization and returns an error. This prevents any one replicated entry on the consumer from having multiple suppliers.

4. The consumer then checks to see if the last update the consumer recorded in its directory is still contained in the supplier's change log. If it is not, then the consumer has no way of knowing what other changes may have occurred on the supplier since the last time the consumer was updated. In this situation, if `orcauto` is enabled, the consumer reinitializes itself from the supplier server; otherwise, you need to reinitialize the consumer. See "Initializing Consumers" on page 344 for more information.

5. If the consumer's last update is still contained in the supplier's change log, then the consumer determines if any changes occurred on the supplier server that must be made to the consumer server's directory. If no changes are required, the consumer terminates synchronization normally. Otherwise, the consumer updates and/or deletes all appropriate entries in its directory tree as indicated by the change log. The consumer sets the appropriate version identification at the same time and then exits synchronization normally.

Machine data

By default, your database actually contains multiple directory trees. One of these trees is used to contain machine data. The machine data tree contains two top-level entries that identify the local server. The first entry uses the `NetscapeMachineData` object class to identify the domain components of the machine on which the server is installed. The second entry uses the `LDAPServer` object class to identify the port on which the LDAP server is listening.

The machine data tree also contains zero or more entries that identify consumer or supplier servers. On the supplier server of an SIR agreement, the machine data tree contains an entry for each consumer server to which the server replicates data. These consumer server entries use object class `LDAPReplica`. On the consumer server of a CIR agreement, the machine data tree contains an entry for each supplier server. The supplier server entries use object class `cirReplicaSource`.

See the *Schema Reference Guide* for information on the `NetscapeMachineData`, `LDAPServer`, `LDAPReplica`, and `cirReplicaSource` object classes.

The suffix for the machine data directory tree is

```
dc=<serverID>, dc=<domain>, dc=<domain_type>
```

For example, if your directory server is running on `directory.airius.com`, then the machine data suffix is

```
dc=directory, dc=airius, dc=com
```

Chapter

14

Managing Referrals

You can use referrals to extend your directory service beyond a single server configuration. This chapter describes how you can use referrals for your directory service. This chapter contains information about referrals in the following sections:

For conceptual information on how you can use referrals in your directory service, see the *Netscape Directory Server Deployment Guide*. For information on starting the server in referral mode, see "Starting the Server in Referral-Only Mode" on page 31.

Understanding Referrals

Referrals are a redirection mechanism supported by the LDAP protocol. There are several reasons why a directory server might return a referral:

- In a replicated environment, the client attempts to modify an entry that is not mastered on the local server. That is, some other directory server supplies the entry to the local server. In this case, the consumer server returns a referral to the client that indicates which server mastered the entry. The client can then follow the referral to the supplier server and attempt the modification operation there.

- If the client requests a directory entry that cannot reside on the local server, then the local server returns a referral based on the value of the `slapd.conf` Referral parameter. The directory server determines whether this kind of a referral should be returned by comparing the DN of the requested directory object against the directory suffixes supported by the local server. If the DN does not match any of the supported suffixes, the directory server returns a referral.

  You can manage this default referral mechanism from the Server Console. See “Setting Default Referral URLs” on page 359 for more information.

- If the client searches an entry, or tries to modify an entry, that contains a smart referral, then a referral is returned based on the LDAP URL contained in the smart referral. See “Creating and Changing Smart Referrals” on page 360 for information.

You can also start the server in referral-only mode. You might want to do this if you’re making configuration changes to the directory server and you want all clients to be referred to another server for the duration. For information on how to do this, see “Starting the Server in Referral-Only Mode” on page 31.

For more information on how referrals are used by LDAP clients and servers, and for information on the reasons why you might want to use referrals, see the *Planning Referrals* chapter in the *Netscape Directory Server Deployment Manual.*

Setting Default Referral URLs

You can configure the server to use one or more LDAP referrals for client requests that are out of bounds for the directory tree(s) serviced by your directory server. These referrals are returned if no relevant smart referrals can be defined for the server. You can manage the default referral mechanism from the Server Console as follows:

1. On the Directory Server Console, select the Configuration tab.
2. Select the root entry in the navigation tree in the left pane.
3. Select the Settings tab in the right pane.
4. Enter an LDAP URL in the “Referrals to” text box and click OK. For example:

   ```
   ldap://directory.airius.com:389/o=airius.com
   ```

   You can enter multiple referral URLs separated by spaces and in quotes as follows:

```
“ldap://dir1.airius.com:389/o=airius.com” “ldap://dir2.airius.com:389/”
```

   Keep in mind that all special characters within a DN must be properly escaped. Commas must be escaped by two backslashes (\\). For example, if the DN includes `o=Airius Bolivia, S.A.`, then the corresponding DN must be

   ```
   o=airius bolivia\\, S.A.
   ```

Creating and Changing Smart Referrals

Smart referrals allow you to map a directory entry or directory tree to a specific LDAP URL. Thus, if a client requests a directory entry such as `uid=bjensen, ou=people, o=airius.com` you can refer the client to a specific server, or a specific entry on a specific server. As a result, for the above DN you could refer the client to the entry `cn=babs jensen, o=people, l=europe, o=airius.com` on the server `directory.europe.airius.com`.

You create and manage smart referrals through the Directory Server Console (See “Creating Smart Referrals Using the Directory Server Console” on page 360 for information) or by using `ldapmodify` (See “Creating Smart Referrals From the Command-line” on page 362 for information.)

Creating Smart Referrals Using the Directory Server Console

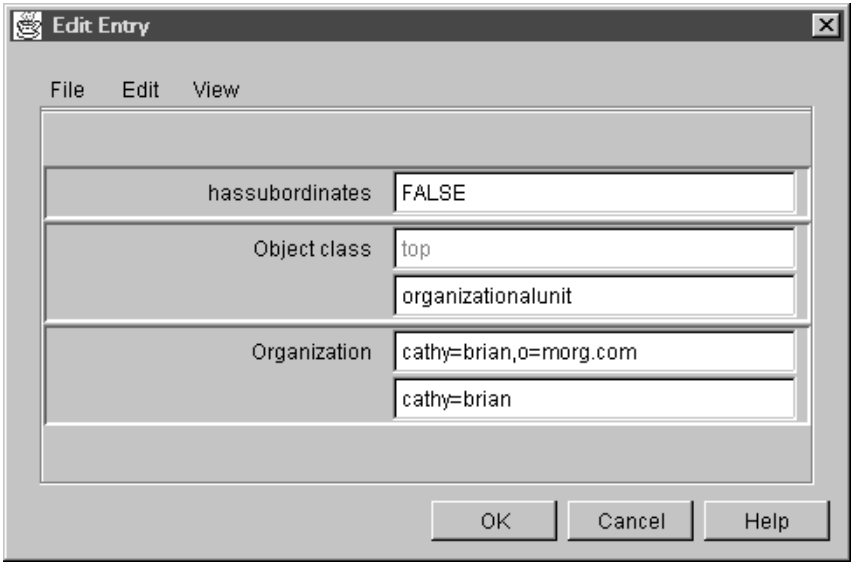
To add a smart referral to an existing entry from the Directory Server Console:

1. On the Directory Server Console, select the Directory tab.
2. Browse through the tree in the navigation pane and select the entry for which you want to add the referral.
3. Right-click the entry and select Open from the pop-up menu. A dialog box or editor that allows you to change the properties of that entry appears.

   If you selected a person, group, or organizational unit entry, the Edit Entry dialog box appears. Click Advanced.

   The property editor dialog box appears containing the object class and attribute values for the entry.

Figure 14.1 Edit Entry (Property Editor) dialog box



4. Right click the cell containing the attribute “Object class” and select Add Value from the pop-up menu.

   The Add Object Class dialog box displays.

5. Select `referral` in the dialog box and click OK.

6. Select Show All Attributes from the View menu.

7. Scroll down the list of attributes to the `ref` attribute.

8. Enter the LDAP URL to which you want to refer queries about this entry in the format `ldap://servername:portnumber/[optional_dn]` in the ref text box. For example,

```
ldap://directory.airius.com:389/cn=csarette,ou=people,o=airius.com
```

   Where `[optional_dn]` is an explicitly specified DN you want the server to return to the requesting client.

If you want the server to use the DN from the original search request instead, enter the LDAP URL in the format:

```
ldap://servername:portnumber
```

Do not include a trailing slash "/" after the URL. For more information on how the server handles referrals and in particular DNs in referrals, see the *Netscape Directory Server Deployment Manual*.

9. Click OK.

Creating Smart Referrals From the Command-line

You use the `ldapmodify` command-line utility to create smart referrals from the command-line. (For more specific information about `ldapmodify`, see “Adding and Modifying Entries Using ldapmodify” on page 243).

To create a smart referral, create the relevant directory entry and add the `Referral` object class. (See “Managing Entries Using the Command-Line Utilities” on page 240 for information.) This object class allows a single attribute: `ref`. The `ref` attribute is expected to contain an LDAP URL.

For example, to return a smart referral for the existing entry `uid=bjensen, ou=people, o=airius.com`, add the following information to the entry:

```
objectclass: referral
ref:
ldap://directory.europe.airius.com/cn=babs%20jensen,ou=people,l=europe,
o=airius.com
```

Note Any information after a space in an LDAP URLs is ignored by the server. For this reason, you must use `%20` instead of spaces in any LDAP URL you intend to use as a referral.

To add the entry `uid=ssarette, ou=people, o=airius.com` with a referral to `directory.europe.airius.com`, you would include the following in your LDIF file before importing:

```
dn: uid=ssarette, ou=people, o=airius.com
objectclass: top
objectclass: person
objectclass: organizationalperson
objectclass: inetOrgPerson
objectclass: referral
  cn: somi sarette
  sn: sarette
  uid: ssarette
  ref:
ldap://directory.europe.airius.com/cn=somi%20sarette,ou=people,l=europe,o=airius.com
```

Use the `-M` parameter with `ldapmodify` to cause the server to not return the smart referral, but treat the entry as a regular entry. For information on the `-M` parameter, see “Additional ldapmodify Parameters” on page 246.

For more information on smart referrals, see Chapter 8 of the *Netscape Directory Server Deployment Manual*. For information on creating entries, see Chapter 9, “Managing Directory Entries,” and for information on LDAP URLs, see Appendix A, “LDAP URLs.”

Chapter

15

NT Directory Synchronization

The Netscape Directory Server NT Synchronization service allows you to synchronize the entries in your Windows NT directory with the entries in your Netscape Directory Server directory. Both NT users and NT groups can be synchronized. As entries are created, modified, or deleted in one directory, the synchronization service makes the corresponding change to the other directory.

Directory synchronization occurs in both directions: NT to directory server and directory server to NT. All NT account information, including passwords, can be synchronized. There are no requirements as to which environment (the NT domain or the directory server) you use to master NT directory data. However, you are recommended to master directory data in just one environment, because this will simplify the administration of your NT accounts. By strictly identifying where and how NT account information is changed, you greatly reduce the chance of conflicts and errors in the management of this information.

The NT Synchronization Service requires the following versions of the directory server:

- If you want to use SSL, Netscape Directory Server 3.0 or higher.
- If you do not want to use SSL, Netscape Directory Server 3.1 or higher.

This chapter contains information about using the NT Synchronization Service in the following sections:

The Synchronization Service

NT directory synchronization is performed using the Netscape directory synchronization service. This service can be configured to automatically synchronize your directories based on a schedule that you supply.

Note A Netscape directory subtree can be managed by one and only one synchronization service; synchronization services from different NT hosts must manage account information in different branches of the directory tree. Moreover, a synchronization service on an NT host can manage multiple subtrees: one for user information and one for group information.

Synchronization: NT to Directory Server

Each NT domain can be synchronized with one and only one directory server. If you are using replication in your directory service, make sure that the directory server that you synchronize with is a supplier server; do not synchronize with a consumer server.

To perform synchronization, the synchronization service examines the local NT directory for changes and then transfers these changes to the directory server using an LDAPS (LDAP over SSL) or LDAP connection.

Note You can not set up two synchronization services so that two separate NT domains are synchronized to a single directory subtree. The directory server includes a plug-in that prevents this from happening. This is prevented because unique user IDs are ensured within an NT domain, but not within multiple NT

domains in the same network. Therefore, if you were to synchronize multiple NT domains to a single directory subtree, you would run the risk of DN conflicts.

How NT Directory Changes Are Discovered

The synchronization service finds changes in the NT directory by using two methods:

- With the exception of passwords, all changes to NT user and group accounts are discovered by examining the Security Accounts Manager (SAM) database. Changes made to this database since the last synchronization are replayed in the directory server.

- Changes to user passwords are captured by a special DLL, which traps and then stores the password changes in encrypted form in a private area. On each synchronization schedule, the synchronization service first examines the SAM file for changes, and then checks this private area for passwords to be synchronized. Once the passwords have been synchronized, the passwords are deleted from the DLL's private area. It is possible to disable NT to directory server password synchronization.

  NT passwords can be synchronized to the directory only if a change is made to the password. If any NT user accounts exist when you first install the synchronization service, then the passwords for those user accounts can not be synchronized until they are changed. This is because the synchronization service cannot determine what a user's password is once it has been encrypted and stored in the NT domain.

  Once the synchronization service has been installed, the machine rebooted, and NT-to-Directory Server password synchronization enabled, changed passwords are captured even if the synchronization service is not running. If the synchronization service is running, the passwords are sent to the directory server at the next scheduled or manually-initiated synchronization.

All NT directory changes can be transferred to the directory server using LDAPS (LDAP over SSL) which ensures the privacy of the NT user and group information. This is the recommended configuration for the synchronization service, although you can configure the synchronization service such that it does not use LDAPS for directory server communications.

Creating User Entries

When you create an NT user account, the synchronization service can automatically create a corresponding entry on the directory server. The new entry is created in the subtree that you identify when you configure the synchronization service.

The new directory server entry is created using the `inetOrgPerson` and `NTUser` object classes. `NTUser` attributes are set as described for the NTUser object class definition in the *Netscape Directory Server Schema Reference* manual.

In addition, the following `inetOrgPerson` attributes are set:

- The common name (`CN`) attribute is set to the NT user account's username field. In addition, if a full name is set on the NT user account, then this value is set as a second `cn` attribute value on the directory server entry.
- The value set for the NT account's password is captured by the synchronization service DLL and transferred to the directory server over LDAPS or LDAP. Password synchronization can be disabled. If it is disabled, then the corresponding directory server entry is created without a password. For information on disabling password synchronization between NT and the directory server, see "Configuring Service Settings" on page 379.
- The `description` attribute is set to the NT user account's comment field.
- The `uid` attribute is also set to the NT user name.

In addition, if the synchronization service has been told to automatically create messaging server accounts, then the `mailRecipient` and `nsLicenseUser` object classes are set on the new directory server account and the following attribute values are set on the entry:

- The `nsLicensedFor` attribute is set given a value of `mail`
- The `mailHost` attribute is set to the value defined for the synchronization service configuration tool's "Create email addresses using the suffix" field in the Account Details tab (see "Configuring Account Details" on page 384 for details).

- The `mail` attribute is set to the string `<mail prefix>@<mailHost>`, where `<mail prefix>` is determined by the value you select on the "Create email addresses using the prefix" field in the Account Details tab. That is, if the `mailHost` attribute is set to `airius.com` and Create email address is set such that the NT user name is used, and the NT user name is `bjensen`, then the `mail` attribute is set to `bjensen@airius.com`.
- The `mailDeliveryOption` attribute is set to `mailbox`

Creating Group Entries

When you create an NT group, the synchronization service can automatically create a corresponding group entry on the directory server. The new group is created in the subtree that you identify when you configure the synchronization service.

The new directory server group entry is created using the `groupOfUniqueNames` and `NTGroup` object classes. `NTGroup` attributes are set as described for the NTGroup object class definition in the *Netscape Directory Server Schema Reference* manual. Specifically, the `ntGroupDomainID` attribute is set with the following value:

```
ntGroupDomainID: <NT domain name>:<NT Group Name>
```

The `ntGroupType` attribute is also set with a value of `local` or `global`.

In addition, the following `groupOfUniqueNames` attributes are set:

- The common name (`cn`) attribute is set to the NT Group name value
- A `uniqueMember` attribute is set for each NT group member
- The `description` attribute is set to the NT description field value

Initially Creating Entries

When you first start the synchronization service, the service does not automatically add any existing NT users to the directory server. To have existing NT users added to the directory server, use the "Add all users and groups" button in the Synchronization Schedule tab.

This causes the synchronization service to add every currently existing user and group to the directory server. The new directory server entries are created as described in "Creating User Entries" on page 371.

Synchronization: Directory Server to NT

Each directory server can communicate with multiple synchronization services. Consequently, the directory server tree should be structured such that all of the NT user entries from a given NT domain are collected within a single directory server subtree. If the directory server is synchronizing with multiple NT domains, then a separate directory server subtree should be used for each NT domain.

How Synchronization Occurs

The directory server uses a LDAP SSL (LDAPS) or LDAP connection to communicate with the NT synchronization service. The directory server uses this connection to:

- Validate any proposed changes to `NTUser` or `NTGroup` information. To validate proposed changes, the directory server uses the non-LDAP connection to check the proposed change against the NT directory rules defined for the domain. If a change is vetoed by the domain, then the LDAP modification operation is rejected by the directory server.
- Transmit `NTUser` and `NTGroup` changes directly to NT for inclusion into the NT domain. This transmission occurs as soon as the change has been made on the directory server. This means that directory server to synchronization service communications do not occur over LDAP.

Because of the importance of this non-LDAP connection, the synchronization service cannot be started if the corresponding directory server is not listening on the port defined for this communication.

Note Changes made to the directory server can only be synchronized to NT if those changes are made over LDAP (that is, if they are made using any LDAP client such as the gateway or the various LDAP command line tools). However, if NT users or groups are created using ldif2db, then those entries will never be discovered by the synchronization service and they will never be synchronized with the NT directory.

Creating User Entries

The NT synchronization service can create an NT user account on the local NT host if the following conditions are true:

- A directory server user entry is created in the subtree being monitored by the synchronization service, or the appropriate attributes are added to an existing directory entry in the subtree being monitored by the synchronization service (see “Associating an Existing Directory User with an NT User Account” on page 374 for details).
- The new or modified directory server entry uses the `NTUser` object class.
- The synchronization service is configured to synchronize user entries. See “Configuring Directory Server Settings” on page 380 for information on turning user entry synchronization on and off.

In this situation, the synchronization service creates the new user based on the following information:

- The NT domain and NT user name is identified based on the information stored in the `ntUserDomainID` attribute.
- The common name (`cn`) attribute is used for the NT user account’s full name field.
- The `description` attribute is used for the NT user account’s comment field.
- The NT account’s password is set to the initial password value of the directory server user entry. (If password synchronization is turned off, then subsequent changes to the directory server user entry’s password are not synchronized to NT. For information on disabling directory server to NT password synchronization, see “Configuring Directory Server Settings” on page 380.)

Creating Group Entries

The NT synchronization service will create an NT group on the local NT host if the following conditions are true:

- A directory server group entry is created in the subtree being monitored by the synchronization service, or the appropriate attributes are added to an existing directory server group entry in the subtree being monitored by the synchronization service (see “Associating an Existing Directory Group with an NT Group” on page 375 for details).
- The new or modified directory server entry uses the `NTGroup` object class.
- The synchronization service is configured to synchronize group entries. See “Configuring Directory Server Settings” on page 380 for details.

In this situation, the synchronization service creates the new group based on the following information:

- The NT domain and NT group name is identified based on the information stored in the `ntGroupDomainID` attribute.
- The common name (`cn`) attribute is used for the NT group’s Groupname field.
- The `description` attribute is used for the NT group’s description field.
- The group is created as an NT local or NT global group depending on the value of the `ntGroupType` attribute.

Creating Duplicate Entries

If an entry is created in the directory server, and the NT user account identified by the `ntUserDomainID` attribute already exists, then the synchronization service’s behavior is dependent upon the directory server entry’s `ntUserCreateNewAccount` attribute. Similarly, if a group is created in the directory server, and the NT group account identified by the `ntGroupDomainID` attribute already exists, then the synchronization service’s behavior is determined by the `ntGroupCreateNewAccount` attribute.

If the `ntUserCreateNewAccount` or `ntGroupCreateNewAccount` attribute does not exist on the entry or if this attribute is set to `false`, then the synchronization service will attempt to modify the existing NT group or user account with the common name and description information stored on the directory server entry.

If the `ntUserCreateNewAccount` or `ntGroupCreateNewAccount` attribute is set to `true`, then the synchronization service will report an error indicating that it attempted to create the new group or user account but that it already existed.

Deleting Entries

The NT synchronization service will delete an NT group or user account if the following conditions are true:

- A directory server entry is deleted from the subtree being monitored by the synchronization service, or the directory server entry is dissassociated from the NT account (see “Dissassociating a Directory User or Group from an NT User or Group” on page 376 for details).
- The directory server entry included the `ntUserDeleteAccount` or `ntUserDeleteGroup` attribute, and this attribute value was set to `true`.
- User and group synchronization is turned on. For information on turning user and group synchronization on and off, see “Configuring Directory Server Settings” on page 380.

Modifying Entries

The synchronization service can modify a user or group account on the local NT host any time the corresponding directory server entry is modified. For details on how this synchronization process occurs, see “How Synchronization Occurs” on page 370.

Associating an Existing Directory User with an NT User Account

If you have an existing directory (LDAP) user entry, and that entry resides in the directory subtree that the synchronization service is monitoring, you can associate the entry with a new or existing NT user account.

You do this by adding the ntUser object class and the required ntUserDomainID attribute to the entry. If the NT user account does not currently exist, use the ntUserCreateNewAccount attribute to cause the synchronization service to create the NT user account for you.

For example, the following LDIF statements associates the existing directory user entry with the `rhunt` user ID in the `CHURCHFIELD` NT domain. The description and ntUserDeleteAccount attributes that are set in the LDIF are optional. If that NT user does not exist, the synchronization service will create it:

```
dn: uid=rhunt, ou=people, o=Airius.com
changetype: modify
add: objectclass
objectclass: ntUser
-
add:ntUserDomainID
ntUserDomainID: CHURCHFIELD:rhunt
-
add: ntUserCreateNewAccount
ntUserCreateNewAccount: true
-
add: description
description: a new NT user
-
add:ntUserDeleteAccount
ntUserDeleteAccount: true
```

Associating an Existing Directory Group with an NT Group

If you have an existing directory (LDAP) group entry, and that entry resides in the directory subtree that the synchronization service is monitoring, you can associate the entry with a new or existing NT group account.

You do this by adding the ntGroup object class and the required ntGroupDomainID attribute to the entry. If the NT group account does not currently exist, use the ntGroupCreateNewGroup attribute to cause the synchronization service to create the NT group account for you.

For example, the following LDIF statements associates the existing directory group entry with the `NT PD Managers` group in the `CHURCHFIELD` NT domain. The description, ntGroupDeleteGroup, and ntGroupType attributes that are set in the LDIF are optional. If that NT group does not exist, the synchronization service will create it:

```
dn: cn=PD Managers, ou=groups, o=Airius.com
changetype: modify
add: objectclass
objectclass: ntGroup
-
add:ntGroupDomainID
ntGroupDomainID: CHURCHFIELD:NT PD Managers
-
add: ntGroupType
ntGroupType: local
-
add: ntGroupCreateNewGroup
ntGroupCreateNewGroup: true
-
add: description
description: a new NT group
-
add:ntGroupDeleteGroup
ntGroupDeleteGroup: true
```

Dissassociating a Directory User or Group from an NT User or Group

You can break the association between a directory (LDAP) user or group and a corresponding NT entry without deleting either entry. You do this by deleting the ntUser or ntGroup object class from the entry and all corresponding attributes.

For example, the following LDIF deletes the ntUser object class from the directory entry and then deletes all the attributes that are allowed by that object class. Doing so causes the synchronization service to no longer synchronize changes made to the directory entry:

```
dn: uid=rhunt, ou=people, o=Airius.com
changetype: modify
delete: objectclass
objectclass: ntUser
-
delete:ntUserDomainID
-
delete: ntUserCreateNewAccount
-
delete: description
description: a new NT user
-
delete:ntUserDeleteAccount
```

Concurrently Changing Directory Server and NT Account Values

Because NT synchronization can be configured to synchronize in two directions, NT to directory server and directory server to NT, there is a potential for losing data. This can happen if you change corresponding entries in both directories before synchronization can occur. For example, if you change an NT user account's comment field and you also change the corresponding directory server entry's description field before the synchronization service can transfer the comment field changes, then you will lose the change that was made first.

The window of opportunity for losing data is driven entirely by the schedule that you set up for NT to directory server synchronization. This is because the directory server to NT synchronization occurs as soon as changes are made on

the directory server. If you make a change on the NT domain, and then make a conflicting change in the directory server before NT to directory server synchronization happens, then the changes to the NT domain will be lost.

Make a habit out of changing NT values in only one of the two directories. This will reduce potential confusion and help to avoid any problems that might occur because of conflicting changes in the two directories.

The Synchronization Configuration Tool

You configure and control directory synchronization using the directory server NT synchronization configuration tool.

This tool is a native Windows NT application that you use to:

- Start and stop the synchronization service
- Identify the directory server with which synchronization should occur
- Identify the directory trees with which NT users and groups are synched
- Identify the NT domain to synchronize
- Identify the ports on which synchronization occurs
- Identify whether SSL is used to communicate with the directory, and if so the certificate database that the synchronization service should use for SSL communications with the directory server (the use of SSL is recommended)
- Schedule synchronization
- Examine synchronization status
- Optionally sets messaging server account details (for more information, see “Configuring Account Details” on page 384)
- Optionally disable/enable synchronization of groups, users, and passwords in one or both directions (NT to directory server, and directory server to NT).

This tool is installed with the NT synchronization service. You can install the NT synchronization service when you install the directory server, or you can install it after the directory server has been installed.

Also, the directory server does not have to be installed on the same physical host as the NT synchronization service; the two can exist on entirely different machines. Additionally, the directory server does not have to be running under NT; the synchronization service will work with directory servers running under Unix operating systems.

About the OK, Cancel, Apply, and Help Buttons

The configuration tool contains four standard buttons:

- Apply—Applies any changes made in the synchronization service settings since the time the tool was opened, or since the time of the last apply.
- Cancel—Cancels any changes that have not been applied and closes the configuration tool window.
- OK—Applies any changes that have not previously been applied, and then closes the configuration tool window.
- Help—Displays online help about the tab that you are currently looking at.

Configuring Synchronization

To set up NT synchronization, you must:

- Use the Service Settings tab to configure the synchronization service to synchronize the correct NT domain, and to use the appropriate port, log file, and (optionally) certificate database. You can optionally use this tab to turn off NT to directory server synchronization completely or just turn on and off synchronization of users, groups, and passwords individually.
- Use the Directory Server Settings tab to identify the directory server and directory subtree with which synchronization is to occur. You can optionally use this tab to turn off directory server to NT synchronization completely or just turn on and off synchronization of users, groups, and passwords individually.

If you are using replication in your directory service, make sure that the directory server that you synchronize with is a supplier server; do not synchronize with a consumer server.

- If you are allowing NT to directory server synchronization, you should set the synchronization schedule using the Synchronization Schedule tab.
- If you are allowing directory server to NT synchronization, make sure the synchronization plug-ins on the directory server are turned on. You do this from the Directory Server Console. For information, see “Enabling and Disabling Plug-Ins From the Server Console” on page 87.
- Determine where the Directory Server to synchronization service connection uses SSL or not. If you want to use SSL, the synchronization service must be configured to use SSL and the directory server must be configured to communicate with the synchronizations service over SSL. For information on how to configure the directory server to communicate with the synchronization service over SSL, see “Creating Certificate Databases for LDAP Clients” on page 313.

Configuring Service Settings

Use the Service Settings tab to identify the following about the synchronization service:

- The NT domain, or the Primary Domain Controller (PDC) that controls the NT domain, that will be synchronized. If the synchronization service is not running on the PDC, then additional setup must be performed to cause the service to login to the domain with an account that has domain privileges. See the *Netscape Directory Server Installation Guide* for details.
- The local port which the synchronization configuration tool uses to communicate with the synchronization service. By default, this is 5007. When the Synchronization Configuration Tool starts up, it opens a connection to the synchronization service on this port and maintains this connection until the tool exits.
- The location of the synchronization service event log file. For example, “c:\Netscape\Server\dssynch\synch-log.txt.” This log file is used by the synchronization service to record significant events and problems. Each time a user or group is added, deleted, modified, or renamed in the NT domain, the synchronization service records the event to this file.

- Whether SSL is used for synchronization. If yes (this is recommended), you also specify the path and filenames of the security certificate database file. For information on how to not use SSL for synchronization, see "Turning Off SSL for the Synchronization Service" on page 386.

  For information on how to obtain a certificate database for your NT Synchronization Service, see "Creating Certificate Databases for LDAP Clients" on page 313. You do not have to obtain a client certificate for this database; you only have to trust the CA used by your directory server, so this database must have the CA's certificate and this certificate must be accepted for certifying network sites.

- If you do not want to synchronize NT groups to the directory server, set the checkbox on the Service Settings tab.

- If you do not want to synchronize NT users to the directory server, set the checkbox on the Service Settings tab.

- If you do not want to synchronize NT passwords to the directory server, set the checkbox on the Service Settings tab.

If you want to disable NT to directory server synchronization, click Disable Synchronization from NT to Directory Server.

Configuring Directory Server Settings

Use the Directory Server Settings tab to identify the following information:

- The fully qualified DNS name of the host on which the directory server is running.

- The port number which the Directory Server is using for communications. If the synchronization service is configured for SSL, then the default is 636. Otherwise it is 389. You cannot change this port number when the synchronization service is running.

  LDAPS connections are not maintained for the life of the synchronization service. Instead, LDAPS connections are established at each scheduled synchronization or when the administrator uses the "Synchronize" or "Add All Users & Groups" buttons on the configuration tool. If there is a problem creating the LDAPS connection to the directory server, a dialog box will be raised and a message will be written to the synchronization log file. Such

problems are often caused by misconfiguration of the synchronization service. See “Troubleshooting Errors at Synchronization Time” on page 387 for information on these misconfigurations.

- The distinguished name and password that the synchronization service should use to bind to the directory server (for example, “`cn=admin, o=airius.com`”). This can either be the Root DN, or it can be a distinguished name that has full read, write, add, delete, search, and compare privileges to the directory server subtree containing the NTUser entries. You are strongly recommended to avoid using the Root DN for normal bind operations such as this.

- The directory base for user entries. This is the directory subtree where the synchronization service will create, modify, and delete user entries. The default is `ou=people, o=<suffix>`. That is, if your directory’s suffix is “`o=airius.com`”, then by default all NT people entries are placed under `ou=people, o=airius.com`.

- The directory base for group entries. This is the directory subtree where the synchronization service will create, modify, and delete group entries. The default is `ou=groups, o=<suffix>`. That is, if your directory’s suffix is “`o=airius.com`”, then by default all NT group entries are placed under `ou=groups, o=airius.com`.

Note If the name of the directory subtree you want to use as the directory base for either users or groups contains a comma, you must escape the comma with a backslash (\) when you enter the value in the directory base field. For example, to use the Airius Bolivia, S.A. subtree as the directory base, you would enter `Airius Bolivia\, S.A.` in the directory base field.

- The directory tree in which you want to enforce uniqueness in the UID. Netscape servers require that all person entries in the directory have a unique `UID` attribute. Most Netscape servers are configured to enforce this uniqueness in the entire directory tree (that is, from the directory suffix down). If Netscape servers are managing users in areas of the directory tree different from the area the synchronization service is managing, then you should use your directory suffix for this field. Otherwise, simply enter the same DN you entered for the directory base for user entries.

  If you leave this field blank, the synchronization service will not enforce UID uniqueness. This is acceptable so long as the directory server itself is enforcing UID uniqueness. It does this through a server plug-in which is turned on by default.

- The port number on which the directory server's synchronization plug-in accepts non-LDAP connections. Default is 5009. When the synchronization service starts up, it establishes a connection to this port, and this is maintained while the synchronization service is running. When these connections are established, a message is written to the synchronization log file. Absence of these messages from the logfile is a good indicator of a problem to come (another indication is if the "Synchronize" and "Add all Users/Groups" buttons are grayed out).

  Synchronization from the Directory Server to NT cannot occur if these connections are not fully established. This sometimes occurs due to misconfiguration of the synchronization service. See "Troubleshooting Errors at Synchronization Time" on page 387 for information on these misconfigurations.

- If you want to disable directory server to NT synchronization, click "Disable Synchronization from Directory Server to NT."

- If you want to disable group synchronization from the directory server to NT, click "Disable Group Synchronization from Directory Server."

- If you want to disable user synchronization from the directory server to NT, click "Disable User Synchronization from Directory Server."

- If you want to disable all password synchronization from the directory server to NT, click "Disable Password Synchronization from Directory Server."

If the Selected UID is Not Unique

If the synchronization service attempts to create a directory entry, and the proposed UID that the synchronization service wants to use is not unique, then the synchronization service:

1. creates the directory entry

2. issues a warning message to the NT Event Log and to the synchronization service's log file (for information on this log file, see "Configuring Service Settings" on page 379)

Scheduling Synchronization

Use the Synchronization Schedule tab to schedule NT to directory server synchronization.

There is no scheduling area for directory server to NT synchronization, because that synchronization always occurs as soon as relevant directory server data is changed.

You use the following two fields to schedule synchronization:

- Synchronize every—This field allows you to specify the interval between synchronization startup times. Units are selectable between minutes and hours. For example, if you select 30 minutes for this field, then synchronization will begin every 30 minutes.

- Start at—This field allows you to specify the time when the synchronization cycle begins. For example, if you specify a time of 1:15 and then you specify an interval of 15 minutes in the Synchronize every field, then synchronization will occur at 1:15, 1:30, 1:45, 2:00, 2:15, and so forth.

  Changing the start time does not cause the synchronization tool to wait for that time to begin synchronization; this field is used only for calculating the next synchronization event. For example, suppose the Start at field is currently set to 1:00 and the Synchronize every field is set to 15 minutes. If the current time is 2:20, then the next synchronization is scheduled for 2:30. Now suppose you change the Start at field to 1:40. Then synchronization will occur at 1:40, 1:55, 2:10, 2:25, and so forth. Therefore, the next synchronization will occur at 2:25 rather than the original 2:30.

The next scheduled synchronization event for each direction is always shown in the “Next synchronization at” field.

Manually Performing Synchronization

While synchronization will always occur based on the schedule set in the synchronization configuration tool, you can manually perform synchronization if you have an immediate need for a synchronization to occur.

To manually perform synchronization, go to the Status tab in the synchronization configuration tool, and click the “Synchronize” button.

The synchronization schedule you have set in the configuration tool is unaffected by this manual synchronization. That is, if a synchronization is scheduled for 1:30 and you perform a synchronization at 1:25, then the 1:30 synchronization will still occur.

Configuring Account Details

Use the Account Details tab to indicate whether the synchronization service should create mail accounts on the directory server when creating new ntUser entries on the directory server.

To cause mail accounts to be created, click "Automatically create Messaging Server accounts for new Directory Server users."

You must then identify the following information:

- the host name of the messaging server for which the new accounts are to be created. In the "Create email addresses using the suffix" field, enter the full DNS host name on which the messaging server is running. For example, `mail.airius.com`.
- The method by which the messaging server user name is generated when the mail account is created. This value must be unique within the subtree managed by the synchronization service.

  The safest algorithm is also the default—the NT user name is used. This is the safest method of mail address creation because the NT user name must be unique within the NT domain managed by the local PDC.

Surname-based NT Accounts

Some cultures commonly begin their names with their surnames. If your NT domain is populated with names such as these, then you must configure the synchronization service with this information so that it can determine how to populate the `surname` and `givenname` attributes for the NT user entries. To do this, go the Account Details tab in the synchronization configuration tool and select "NT account full name begins with surname."

Starting and Stopping the Synchronization Service

Use the Status tab to start and stop the synchronization service.

The service is not running when it is first installed. However, the service is configured to automatically start when your NT system reboots. To reconfigure the service so that it does not start when NT reboots:

1. Go to the NT Control Panel, and select Services.
2. Scroll through the list of services until you see Netscape Directory Synchronization Service. The Startup field is set to `Automatic`.
3. Double-click on Netscape Directory Synchronization Service.
4. Select the Manual radio button, and then click OK.
5. Click Close in the Services control panel.

Checking Synchronization Status

You can use the synchronization configuration tool to check synchronization status. From the Status area, you can determine:

- whether the synchronization service is running
- when synchronization is scheduled to occur for each synchronization direction
- whether the directory server is running

Turning Off SSL for the Synchronization Service

You are strongly recommended to use SSL for NT to directory synchronization because of the sensitive nature of the information that you are synchronizing. However, you may feel that SSL is unnecessary, especially if you are using the synchronization service in a non-production (lab) environment. Do the following to cause the synchronization service to not use SSL when synchronizing with your directory server.

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
2. Select the Settings tab in the right pane.
3. Clear the "Use SSL in NT Synchronization Service" checkbox.
4. Click Save and then restart the server. See "Starting and Stopping the Directory Server" on page 29 for more information.
5. Go to the Service Settings tab in the NT Synchronization Configuration Tool and make sure that "Use SSL" is not selected.
6. In the Directory Server Settings tab, make sure the LDAP port is not set to 636 (it should probably be 389).
7. Save your changed settings and restart the synchronization service.

Troubleshooting Errors at Synchronization Time

If your synchronization service is not properly configured, synchronization does not occur and a message box is raised by the configuration tool indicating the error.

The message box indicates that the directory base DNs and/or other configuration attributes are not correct. Ensure that the directory base DNs, and the administrator DN and password are correct. Also verify that the port numbers used for LDAP/LDAPS and the synchronization plug-ins match on the directory server and the synchronization service.

If the message box indicates error 81, then the synchronization service and/or the directory server have not been properly configured for SSL communications. Examine the directory server access log file to see if the connection attempt was received by the directory server. You may also find helpful messages in the directory server's error log file.

To narrow down the source of the misconfiguration, try to establish an LDAPS connection to the directory server using Netscape Communicator. If this connection attempt fails, check all values (port number, host name, search base, and so forth) to see if any of these are the problem. If all else fails, reconfigure the directory server with a new certificate.

Note A common problem is to fail to trust your certificate authority when configuring synchronization service's certificate database. For information on how to create a certificate database, see "Creating Certificate Databases for LDAP Clients" on page 313.

If the Communicator connection is successful, it is likely that the misconfiguration is on the synchronization service side. Recheck all configuration values and examine the synchronization log file for error messages. To help you determine whether the problem is with SSL configuration or general synchronization service configuration, turn off SSL for synchronization and make sure that synchronization is working. Wait until everything is working before you try to configure the synchronization service with SSL.

Chapter

16

Managing SNMP

You can monitor your directory server in real time using the Simple Network Management Protocol (SNMP).

This chapter contains the following topics:

Understanding SNMP

SNMP is a protocol used to exchange data about network activity. With SNMP, data travels between a managed device and a *network management station* (NMS) where users remotely manage the network.

A managed device is anything that runs SNMP, such as hosts, routers, and your directory server.

An NMS is usually a powerful workstation with one or more network management applications installed. A *network management application* graphically shows information about managed devices (which device is up or down, which and how many error messages were received, and so on).

Information is transferred between the NMS and the managed device through the use of two types of agents: the *subagent* and the *master agent*. The subagent gathers information about the managed device and passes the information to the master agent. The Netscape Directory Server has a subagent. The master agent exchanges information between the various subagents and the NMS. The master agent runs on the same host machine as the subagents it talks to.

You can have multiple subagents installed on a host machine. For example, if you have the directory server, the Enterprise Server, and the Messaging Server all installed on the same host, the subagents for each of these servers communicates with the same master agent. In the Windows NT environment, the master agent is the SNMP service provided by the Windows NT operating system. In the Unix environment, the master agent is installed with the Netscape Administration Server.

Values for variables that can be queried are kept on the managed device and reported to the NMS as necessary. Each variable is known as a managed object, which is anything the agent can access and send to the NMS. All managed objects are defined in a *management information base* (MIB), which is a database with a tree-like hierarchy. The top level of the hierarchy contains the most general information about the network. Each branch underneath is more specific and deals with separate network areas.

SNMP Overview

SNMP exchanges network information in the form of *protocol data units* (PDUs). PDUs contain information about variables stored on the managed device. These variables, also known as managed objects, have values and titles that are reported to the NMS as necessary. Communication between an NMS and a managed device takes place in one of two forms. These forms are described in the following sections:

- "NMS-Initiated Communication" on page 391
- "Managed Device-Initiated Communication" on page 391

NMS-Initiated Communication

NMS-initiated communication is the most common type of communication between an NMS and a managed device. In this type of communication, the NMS either requests information from the managed device or changes the value of a variable stored on the managed device.

These are the steps that make up an NMS-initiated SNMP session:

1. The NMS determines which managed devices and objects need to be monitored.
2. The NMS sends a PDU to the managed device's subagent through the master agent. This PDU either requests information from the managed device or tells the subagent to change the values for variables stored on the managed device.
3. The subagent for the managed device receives the PDU from the master agent.
4. If the PDU from the NMS is a request for information about variables, the subagent gives information to the master agent and the master agent sends it back to the NMS in the form of another PDU. The NMS then displays the information textually or graphically.

   If the PDU from the NMS requests that the subagent set variable values, the subagent sets these values.

Managed Device-Initiated Communication

This type of communication occurs when the managed device needs to inform the NMS of an event that has occurred. A managed device initiates communication with an NMS to inform the NMS of a shut down or start up. Communication initiated by a managed device is also known as a "trap." The directory server sends a trap to the NMS whenever the directory server starts or stops.

These are the steps that make up a managed device-initiated SNMP session:

1. An event occurs on the managed device.
2. The subagent informs the master agent of the event.

3. The master agent sends a PDU to the NMS to inform the NMS of the event.

4. The NMS displays the information textually or graphically.

The Directory Server MIB

Each Netscape server has its own MIB. The directory server's MIB is a file called `netscape-ldap.mib`. This MIB contains definitions for variables pertaining to network management for the directory server. These variables are known as managed objects. Using the directory server MIB and network management software, such as HP OpenView, you can monitor your directory server like all other managed devices on your network.

The directory server MIB has an object identifier of `iso.org.dod.internet.private.enterprises.netscape.nsldap` (that is, `nsldap OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.1450.7 })` and is located in the `<NSHOME>/plugins/snmp` directory.

You can see administrative information about your directory server and monitor the server in real time using the directory server MIB. The directory server MIB is broken into three distinct tables of managed objects:

- The Operations Table
- The Entries Table
- The Interaction Table

Note Before you can use the directory server's MIB, you must compile it along with the MIBs that you will find in the following location:

`<NSHOME>/plugins/snmp/mibs`

For information on how to compile MIBs, see your SNMP product documentation.

The Operations Table

The Operations Table provides statistical information about directory server access, operations, and errors. Table 16.1 describes the managed objects stored in the Operations Table of the `netscape-ldap.mib` file.

Table 16.1 netscape-ldap.mib Operations Table managed objects and descriptions

Managed object	Description
dsAnonymousBinds	The number of anonymous binds to the directory since server startup.
dsUnauthBinds	The number of unauthenticated binds to the directory since server startup.
dsSimpleAuthBinds	The number of binds to the directory server that were established using a simple authentication method such as password protection since server startup.
dsStrongAuthBinds	The number of binds to the directory server that were established using a strong authentication method such as SSL or a SASL mechanism such as Kerberos since server startup.
dsBindSecurityErrors	The number of bind requests that have been rejected by the directory server due to authentication failures or invalid credentials since server startup.
dsInOps	The number of operations forwarded to this directory server from another directory server since server startup.
dsReadOps	The number of read operations serviced by this directory server since application start. The value of this object will always be 0 because LDAP implements read operations indirectly via the search operation.
dsCompareOps	The number of compare operations serviced by this directory server since server startup.
dsAddEntryOps	The number of add operations serviced by this directory server since server startup.
dsRemoveEntryOps	The number of delete operations serviced by this directory server since server startup.

Table 16.1 netscape-ldap.mib Operations Table managed objects and descriptions

Managed object	Description
dsModifyEntryOps	The number of modify operations serviced by this directory server since server startup.
dsModifyRDNOps	The number of modify RDN operations serviced by this directory server since server startup.
dsListOps	The number of list operations serviced by this directory server since server startup. The value of this object will always be 0 because LDAP implements list operations indirectly via the search operation.
dsSearchOps	The total number of search operations serviced by this directory server since server startup.
dsOneLevelSearchOps	The number of one-level search operations serviced by this directory server since server startup.
dsWholeSubtreeSearchOps	The number of whole subtree search operations serviced by this directory server since server startup.
dsReferrals	The number of referrals returned by this directory server in response to client requests since server startup.
dsChainings	The number of operations forwarded by this directory server to other directory servers since server startup. The value of this object will always be `0`.
dsSecurityErrors	The number of operations forwarded to this directory server that did not meet security requirements.
dsErrors	The number of requests that could not be serviced due to errors (other than security or referral errors). Errors include name errors, update errors, attribute errors, and service errors. Partially serviced requests will not be counted as an error.

The Entries Table

The Entries Table provides statistical information about the contents of the directory server entries. Table 16.2 describes the managed objects stored in the Entries Table of the `netscape-ldap.mib` file.

Table 16.2 netscape-ldap.mib Entries Table managed objects and descriptions

Managed object	Description
dsMasterEntries	The number of directory entries for which this directory server contains the master entry. The value of this object will always be `0`.
dsCopyEntries	The number of directory entries for which this directory server contains a slave copy. The value of this object will always be `0`.
dsCacheEntries	The number of entries cached in the directory server.
dsCacheHits	The number of operations serviced from the locally held cache since application startup.
dsSlaveHits	The number of operations that were serviced from locally held replications (shadow entries). The value of this object will always be `0`.

The Interaction Table

The Interaction Table provides statistical information about the interaction of this directory server with peer directory servers. This table contains statistical information for the last 5 directory servers with which this directory server has attempted to communicate. This table provides useful information about how the interaction with peer directory servers affects the performance of this directory server. Table 16.3 describes the managed objects stored in the Interaction Table of the `netscape-ldab.mib` file.

Table 16.3 netscape-ldap.mib Interaction Table managed objects and descriptions

Managed object	Description
dsIntIndex	Statistical data is kept for the last 5 peer directory servers with which this directory server has attempted to communicate. This object provides a unique identifier used to delimit the information about the interaction with a specific peer directory server.
dsName	The distinguished name of the peer directory server identified by the corresponding `dsIntIndex` object.
dsTimeOfCreation	The amount of time since this directory server first attempted to contact the peer directory server identified by the corresponding `dsIntIndex` object. If this attempt was made before the NMS was initialized, the object will contain a value of `0`.
dsTimeOfLastAttempt	The amount of time since this directory server last attempted to contact the peer directory server identified by the corresponding `dsIntIndex` object. If this attempt was made before the NMS was initialized, the object will contain a value of `0`.
dsTimeOfLastSuccess	The amount of time since this directory server last successfully contacted the peer directory server identified by the corresponding `dsIntIndex` object. If this contact was made before the NMS was initialized, the object will contain a value of `0`.
dsFailuresSinceLastSuccess	The number of times this directory server has failed to contact the peer directory server identified in the corresponding `dsIntIndex` object since the last successful contact.
dsFailures	The total number of times this directory server has failed to contact the peer directory server identified by the corresponding `dsIntIndex` object.

Table 16.3 netscape-ldap.mib Interaction Table managed objects and descriptions

Managed object	Description
dsSuccesses	The total number of times this directory server has successfully contacted the peer directory server identified by the corresponding `dsIntIndex` object.
dsURL	The URL of the peer directory server identified in the corresponding `dsIntIndex` object.

Setting Up SNMP

The steps for configuring SNMP support for your directory server depend on whether your directory server runs on Windows NT or on Unix.

Setting Up SNMP on Windows NT

To set up SNMP support for your directory server on a Windows NT machine:

1. Install the SNMP service on your NT server.

   Refer to your Windows NT operating system documentation for instructions.

2. Enable directory server statistics collection. See “Configuring SNMP for the Directory Server” on page 399 for information.

3. Restart the Windows NT SNMP service.

Setting Up SNMP on Unix

To set up SNMP support for your directory server on a Unix machine:

1. Configure and start the master agent using the administration server interface.

   For information on setting up the Master Agent, refer to *Managing Servers with Netscape Console*.

2. AIX Only. Configure the AIX SNMP Daemon.

   See “Configuring the AIX SNMP Daemon (AIX Only)” on page 398 for information.

3. Enable the directory server subagent.

   See “Configuring SNMP for the Directory Server” on page 399 for information.

4. Start the directory server subagent.

   See “Starting and Stopping the SNMP Subagent on Unix” on page 399 for information.

Configuring the AIX SNMP Daemon (AIX Only)

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

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

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

```
smux 1.3.6.1.4.1.1.1450.7 "" <IP_address> <net_mask>
```

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

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

If you need more information, see your related system documentation for details.

Starting and Stopping the SNMP Subagent on Unix

To start, stop, and restart the SNMP subagent for a directory server running on Unix:

1. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
2. Select the SNMP tab in the right pane.
3. Click Start to start the subagent, click Stop to stop the subagent, or click Restart to restart the subagent.

   Stopping the directory server does not stop the directory subagent. If you want to stop the subagent, you must do so from this tab.

   If you add another server instance and you want the instance to be part of the SNMP network, you must restart the subagent.

Configuring SNMP for the Directory Server

To configure SNMP settings for the directory server from the Directory Server Console:

1. Make sure the directory server is running.
2. On the Directory Server Console, select the Configuration tab and then select the root entry in the navigation tree in the left pane.
3. Select the SNMP tab in the right pane.

4. Select the “Enable Statistics Collection” checkbox to enable directory server statistics collection. Clear the checkbox to disable it.

5. For Unix servers, enter the hostname on which the master agent resides and the port number used to communicate with the master agent in the Master Host and Master Port text boxes.

   The defaults are `localhost` and `199` respectively.

6. Enter a description that uniquely describes the directory server instance in the Description text box.

7. Type the name the company or organization to which the directory server belongs in the Organization text box.

8. Type the location within the company or organization where the directory server resides in the Location text box.

9. Type the email address of the person responsible for maintaining the directory server in the Contact text box.

10. Click Save.

11. Restart the subagent (Unix), or restart the SNMP service (Windows NT). See “Starting and Stopping the SNMP Subagent on Unix” on page 399 for information.

Chapter

17

Configuration Parameters

Directory server runtime activities are controlled using configuration parameters. This chapter details the configuration parameters used with the directory server and includes the following topics:

Changing Configuration Parameter Values

You can change parameter values through the server console. Alternatively, you can change these parameter values by directly editing the `slapd.conf` or `slapd.ldbm.conf` file.

Changing Parameter Values Using the Server Console

You can change most server parameter values from the Directory Server Console. Unlike previous versions of the directory server, there is no longer a single place that you can go in the server console that allows you to view all server parameters from a single form. Instead, individual server parameters are viewed and set in areas of the UI specific to that task. That is, to change parameters related to the access log, you would go to the Configuration|Logs|Access Log tab on the Directory Server Console.

Changing Parameter Values Using slapd.conf

The `slapd.conf` file is a text (UTF-8) file that is read only when the directory server is started. `slapd.conf` contains all of the server parameters that are not related to the server's database. You should not manually edit this file while the server is running because any changes made through the server console cause the server to rewrite the file and may overwrite your manual changes. For a list of these parameters, see Table 17.1 on page 404.

To modify this file:

1. Stop the server.
2. Edit the file with the text editor of your choice.
3. Stop and then restart the directory server.

The location of all of the directory server's configuration files is documented in "Directory Server Configuration Files" on page 36.

slapd.conf File Format

The `slapd.conf` file begins with several `include` statements that include the standard attribute and object class definitions. The remainder of `slapd.conf` consists of a series of general configuration parameters that apply to the directory server as a whole, followed by a database definition that contains information specific to the database.

Note General parameters may be repeated within the database definition. The last instance of any repeated parameter takes precedence over all other duplicated parameters. The only restriction is that all non-database parameters must appear in the file before any database-specific parameters.

Comment lines begin with a pound symbol (#). Blank lines and comment lines are ignored by the directory server. A line beginning with white space is considered a continuation of the previous line.

Note Comments are not preserved if the server rewrites the configuration files.

Entry arguments are separated by white space. If a parameter value contains white space, then it must be enclosed in double quotation marks (for example, “like this”). If a parameter value contains a double quotation mark (“) or a backslash (\), the character must be preceded (escaped) by a backslash character.

Also, file paths contained in the config file must be delimited using a forward slash (/). Backslashes (\) are not supported. For example, an include directive on an NT system should be written as follows:

```
include c:/usr/ns-home/slapd-phonebook/config/slapd.at.conf
```

The format of the `slapd.conf` file is:

```
# comment - slapd.at.conf contains common attribute
# definitions, slapd.oc.conf contains common
# object class definitions.
include /usr/ns-home/slapd-phonebook/config/slapd.at.conf
include /usr/ns-home/slapd-phonebook/config/slapd.oc.conf

# The first parameters apply to the directory server as a whole
<general parameter>
<general parameter>
...
# The dynamicconf parameter that follows includes the file that contains
# the server’s database parameters.
dynamicconf /usr/ns-home/slapd-phonebook/config/slapd.ldbm.conf
```

Changing Parameter Values Using slapd.ldbm.conf

The `slapd.ldbm.conf` file is used to contain the directory server's database parameters. This file is included into `slapd.conf` using the `dynamicconf` parameter. For a list of the database parameters, see Table 17.2 on page 473.

`slapd.ldbm.conf` is a text (UTF-8) file.

General Server Parameters

Table 17.1 describes the server parameters that apply to general directory server operations. They are all contained in the `slapd.conf` file.

Table 17.1 Directory server general parameters

Parameter	Description
Access Log	String specifying the file used to log information about each database access.
Access Log Enable Logging	Boolean specifying whether access logging is on.
Access Log Expiration Time	Integer specifying the maximum age of a log file.
Access Log Expiration Time Unit	Keyword specifying the unit for the Access Log Expiration Time parameter.
Access Log Maximum Disk Space	Integer specifying the maximum amount of disk space that the access logs can use.
Access Log Maximum Log Size	Integer specifying the maximum size of an access log file.
Access Log Maximum Number of Log Files	Integer specifying the total number of access log files that can be in the access log directory.

Table 17.1 Directory server general parameters (Continued)

Parameter	Description
Access Log Minimum Free Disk Space	Integer specifying the minimum amount of free disk space allowed before old log files are deleted.
Access Log Rotation Time	Integer indicating the amount of time between log file rotations.
Access Log Rotation Time Unit	Keyword specifying the units for the Access Log Rotation parameter.
accessloglevel	Reserved for future use.
Account Lockout	Boolean indicating whether users will be locked out of the directory after a given number of failed bind attempts.
Attribute	String associating a syntax with an attribute name. This parameter can only be updated by editing `slapd.conf`; it cannot be edited in the server console.
Audit Log	String specifying the file used to store changes made to each database as well as the machine data area.
Audit Log Enable Logging	Boolean specifying whether audit logging is on.
Audit Log Expiration Time	Integer specifying the maximum age of a log file.
Audit Log Expiration Time Unit	Keyword specifying the unit for the Audit Log Expiration Time parameter.
Audit Log Maximum Disk Space	Integer specifying the maximum amount of disk space that the audit logs can use.
Audit Log Maximum Log Size	Integer specifying the maximum size of a audit log file.
Audit Log Maximum Number of Log Files	Integer specifying the total number of audit log files that can be in the audit log directory.
Audit Log Minimum Free Disk Space	Integer specifying the minimum amount of free disk space allowed before old log files are deleted.
Audit Log Rotation Time	Integer indicating the amount of time between log file rotations.

Table 17.1 Directory server general parameters (Continued)

Parameter	Description
Audit Log Rotation Time Unit	Keyword specifying the units for the Audit Log Rotation parameter.
Certificate and Key Directory	String specifying the path to the SSL directory. This parameter can only be updated by editing `slapd.conf`; it cannot be edited in the server console.
Changelog DB Directory	String specifying the suffix for the change log database.
Changelog Suffix	String displaying the suffix for the change log database.
Check Password Syntax	Boolean indicating whether the password syntax will be checked before the password is saved.
Enable Access Control	Boolean indicating whether access control checking is turned off.
Enable Online Consumer Creation	Indicates whether a server will automatically use online consumer (replica) creation in the event that an inconsistency is detected between the databases on the supplier and the consumer servers.
Enable Superior Object Class Enquoting	Boolean specifying whether object classes in the cn=schema tree will conform to quoting as specified in RFC 2252.
Encrypted Port Number	Integer specifying the TCP/IP port number used for SSL communications.
Encryption Alias	String representing the encryption alias for this server's certificate.
Encryption Ciphers	String specifying the type of encryption supported by this server.
Error Log	String specifying the file used to log error messages generated by the directory server.
Error Log Enable Logging	Boolean specifying whether error logging is on.
Error Log Expiration Time	Integer specifying the maximum age of a log file.
Error Log Expiration Time Unit	Keyword specifying the unit for the Error Log Expiration Time parameter.

Table 17.1 Directory server general parameters (Continued)

Parameter	Description
Error Log Maximum Disk Space	Integer specifying the maximum amount of disk space that the error logs can use.
Error Log Maximum Log Size	Integer specifying the maximum size of an error log file.
Error Log Maximum Number of Log Files	Integer specifying the total number of error log files that can be in the error log directory.
Error Log Minimum Free Disk Space	Integer specifying the minimum amount of free disk space allowed before old log files are deleted.
Error Log Rotation Time	Integer indicating the amount of time between log file rotations.
Error Log Rotation Time Unit	Keyword specifying the units for the Error Log Rotation parameter.
Idle Timeout	Seconds after which idle LDAP client connections are closed.
Instance Directory	String providing the path to the server's installation directory.
IO Block Time Out	Milliseconds after which the connection to a stalled LDAP client that has not made any I/O progress for read or write is closed.
Listen to IP Address	IP address that the directory server listens to. Used on multihomed systems only.
Local User	String indicating the user that the directory server runs as. Used by Unix installations only.
Lockout Duration	Integer representing the amount of time in minutes that users will be locked out of the directory after an account lockout.
Log Buffering	Forces all access log entries to write through the buffer and direct to disk.
Log Level	Integer representing the level at which debugging statements and operation statistics will be logged.
Max Changelog Age	Integer and ID specifying the maximum allowable age of any entry in the change log.
Max Changelog Records	Integer representing the maximum number of records the change log may contain.

Table 17.1 Directory server general parameters (Continued)

Parameter	Description
Maximum File Descriptors	Specifies the number of file descriptors available to the directory server. Not applicable to NT and AIX installations of the directory server.
Maximum Message Size	Maximum size of any add or modification request that can be written to the server over LDAP.
Maximum Password Failures	Integer representing the number failed bind attempts after which a user will be locked out of the directory.
Maximum Threads Per Connection	Maximum number of threads allowed for use by each connection. This parameter can only be updated by editing `slapd.conf`; it cannot be edited in the server console.
nagle	Reserved for future use.
NLS	String that displays the directory where the files to support internationalization are kept.
NT Synchronization Service Enabled	Turns on the NT Synchronization Service server plug-ins.
NT Synchronization Service Port Number	Indicates the port that the directory server will use to for non-LDAP communications with the NT Synchronization Service.
NT Synchronization Service Use SSL	Indicates whether the server will use SSL when communicating with the NT Synchronization Service.
Number of Passwords to Remember	Integer representing the number of passwords the directory server stores in history.
Object Class	List of strings defining a new object class to be added to the database schema. This parameter can only be updated by editing `slapd.conf`; it cannot be edited in the server console.
Password Change	Keyword indicating whether users can change their passwords.
Password Expiration	Boolean indicating whether user passwords will expire after a given number of days.
Password History	Boolean indicating whether users can reuse passwords.
Password Maximum Age	Integer representing the number of days after which user passwords will expire.

Table 17.1 Directory server general parameters (Continued)

Parameter	Description
Password Minimum Age	Integer representing the minimum number of seconds that must pass before a user can change their password.
Password Minimum Length	Integer representing the minimum number of characters that must be used in directory server passwords.
Password Must Change	Keyword indicating whether users must change their passwords when they first bind to the directory server.
Password Storage Scheme	String specifying the type of encryption used for password storage.
Port Number	Integer specifying the TCP/IP port number used for non-SSL communications.
Referral	String specifying an LDAP URL to pass back to a client when `ns-slapd` cannot find a local database to handle a request.
Reserved File Descriptors	Specifies the number of file descriptors reserved by the directory server for non-connection uses. Not applicable to NT and AIX installations of the directory server.
Reset Password Failure Count After	Integer representing the amount of time in minutes after which the password failure counter will be reset.
result_tweak	Reserved for future use.
Root DN	String specifying the distinguished name of an entry that is not subject to access control or administrative limit restrictions for operations on the database.
Root Password	String displaying the current root password.
Root Password Storage Scheme	String displaying the current root password encryption method used for the root password.
Schema Checking	Boolean indicating whether the schema will be enforced during entry insertion or modification.
Security	Boolean specifying whether the server is to use SSL communications.
Send Warning	Integer representing the number days before a user's password is due to expire that the user will be sent a warning message.
Size Limit	Integer specifying the maximum number of entries to return from a search operation.

Table 17.1 Directory server general parameters (Continued)

Parameter	Description
Supplier DN	String specifying the distinguished name used to update local replicated entries.
Supplier Password	String specifying the password the consumer server expects the supplier server to use when binding.
Supplier SSL Clients	String specifying the subject name(s) or the certificate(s) that correspond to the supplier DN defined for the consumer server.
Thread Number	Number of threads obtained by the directory server at startup time. This parameter can only be updated by editing `slapd.conf`; it cannot be edited in the server console.
Time Limit	Integer specifying the maximum number of seconds the directory server will spend performing a search request.
Track Modification Time	Boolean indicating whether `ns-slapd` will maintain modification attributes for entries.
Unlock Account	Boolean indicating whether users will be locked out of the directory until the administrator resets the password after an account lockout.
User-Defined Attributes File	String providing the path to the file containing the user-defined attributes.
User-Defined Object Class File	String providing the path to the file containing the user-defined object classes.

Access Log

Specifies the path and filename of the log used to record each database access. The following information is recorded by default in the log file:

- IP address of the client machine that accessed the database
- operations performed (for example, search, add, modify)
- result of the access (for example, the number of entries returned)

To turn access logging off, leave this parameter blank. For more information on turning access logging off, see “Configuring the Access Log” on page 265.

Default value

<NSHOME>/slapd-<serverID>/logs/access

Valid range

Any valid filename

slapd.conf Syntax

```
accesslog <filename>
```

Example

```
accesslog “/usr/ns-home/slapd-<serverID>/logs/access”
```

Access Log Enable Logging

Turns access logging on and off.

Default value

on

Valid range

on | off

slapd.conf Syntax

```
accesslog-logging-enabled <Boolean>
```

Example

```
accesslog-logging-enabled on
```

Access Log Expiration Time

Specifies the maximum age that a log file is allowed to be before it is deleted. This parameter supplies only the number of units. The units (day, week, month, and so forth) are given by the Access Log Expiration Time Unit parameter.

Default value

1

Valid range

-1 | 1 to 65535

A value of -1 means that an access log will never be deleted due to its age.

slapd.conf Syntax

```
accesslog-logexpirationtime <integer>
```

Example

```
accesslog-logexpirationtime 1
```

Access Log Expiration Time Unit

Specifies the units for Access Log Expiration Time.

Default value

week

Valid range

month | week | day | hour | minute

slapd.conf Syntax

```
accesslog-logexpirationtimeunit <keyword>
```

Example

```
accesslog-logexpirationtimeunit day
```

Access Log Maximum Disk Space

Specifies the maximum amount of disk space in megabytes that the access logs are allowed to consume. If this value is exceeded, the oldest access log is deleted.

When setting a maximum disk space, consider the total number of log files that can be created due to log file rotation. Also remember that there are 3 different log files (access log, audit log, and error log) maintained by the directory server, each of which will consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the access log.

Default value

500

Valid range

-1 | 1 to 65535

A value of -1 means that the disk space allowed to the access log is unlimited in size.

slapd.conf Syntax

```
accesslog-maxlogdiskspace <integer>
```

Example

```
accesslog-maxlogdiskspace 500
```

Access Log Maximum Log Size

Specifies the maximum access log size in megabytes. When this value is reached, the access log is rotated. That is, the server starts writing log information to a new log file. If you set “Access Log Maximum Number of Log Files” to 1, the server ignores this parameter.

Default value

100

Valid range

-1 | 1 to 65535

A value of -1 means the log file is unlimited in size. When setting a maximum log size, consider the total number of log files that can be created due to log file rotation. Also remember that there are 3 different log files (access log, audit log, and error log) maintained by the directory server, each of which will consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the access log.

slapd.conf Syntax

```
accesslog-maxlogsize <integer>
```

Example

```
accesslog-maxlogsize 100
```

Access Log Maximum Number of Log Files

Specifies the total number of access logs that can be contained in the directory where the access log is stored. If you are using log file rotation, then each time the access log is rotated, a new log file is created. When the number of files contained in the access log directory exceeds the value stored on this parameter, then the oldest version of the log file is deleted. Do not set this value to 1. If you do, the server will not rotate the log and it will grow indefinitely.

Default value

10

Valid range

1 to 65535

slapd.conf Syntax

```
accesslog-maxNumOfLogsPerDir <integer>
```

Example

```
accesslog-maxNumOfLogsPerDir 10
```

Access Log Minimum Free Disk Space

Specifies the minimum allowed free disk space in megabytes. When the amount of free disk space falls below the value specified on this parameter, the oldest access log is deleted until enough disk space is freed to satisfy this parameter.

Default value

5

Valid range

1 to 65535

slapd.conf Syntax

```
accesslog-minfreediskspace <integer>
```

Example

```
accesslog-minfreediskspace 5
```

Access Log Rotation Time

Specifies the time between access log file rotations. The access log will be rotated when this time interval is up, regardless of the current size of the access log. This parameter supplies only the number of units. The units (day, week, month, and so forth) are given by the Access Log Rotation Time Unit parameter. If you set “Access Log Maximum Number of Log Files” to 1, the server ignores this parameter.

Default value

1

Valid range

-1 | 1 to 65535

A value of -1 means that the time between access log file rotation is unlimited.

slapd.conf Syntax

```
accesslog-logrotationtime <integer>
```

Example

```
accesslog-logrotationtime 100
```

Access Log Rotation Time Unit

Specifies the units for Access Log Rotation Time.

Default value

day

Valid range

month | week | day | hour | minute

slapd.conf Syntax

```
accesslog-logrotationtimeunit <keyword>
```

Example

```
accesslog-logrotationtimeunit day
```

accessloglevel

Reserved for future use. Do not change or remove. Doing so can have unpredictable results.

Account Lockout

Indicates whether users will be locked out of the directory after a given number of failed bind attempts. By default, users will not be locked out of the directory after a series of failed bind attempts. If you enable account lockout, you can set the number of failed bind attempts after which the user will be locked out using the Maximum Password Failures parameter.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

On

Valid Range

On | Off

slapd.conf Syntax

```
pw_lockout <Boolean>
```

Example

```
pw_lockout off
```

Attribute

Associates a syntax with an attribute name. By default, an attribute is assumed to have syntax `cis`. This parameter also allows you to specify one or more optional alternate names for the attribute.

This parameter is intended to allow the extension of the standard schema when schema checking is turned on.

For details on extending the schema using the Directory Server Console, refer to Chapter 3, “Extending the Directory Schema.”

This parameter is not available from the server console.

Valid range

Possible syntaxes are:

- `bin`—binary
- `ces`—case exact string (case must be matched during comparison)
- `cis`—case ignore string (case is ignored during comparison)
- `tel`—telephone number (identical to `cis`, but blanks and dashes [-] are ignored during comparisons)
- `dn`—distinguished name
- `int`—integer

slapd.conf Syntax

```
attribute <name> [<name2> <syntax>]
```

Example

```
attribute commonName cn cis
```

Audit Log

Specifies the pathname and filename of the log used to record changes made to each database as well as to the machine data area.

Default value

<NSHOME>/slapd-<serverID>/logs/audit

Valid range

Any valid filename

slapd.conf Syntax

```
auditfile <filename>
```

Example

```
auditfile /usr/ns-home/slapd-<serverID>/logs/audit
```

Audit Log Enable Logging

Turns audit logging on and off.

Default value

on

Valid range

on | off

slapd.conf Syntax

```
auditlog-logging-enabled <Boolean>
```

Example

```
auditlog-logging-enabled on
```

Audit Log Expiration Time

Specifies the maximum age that a log file is allowed to be before it is deleted. This parameter supplies only the number of units. The units (day, week, month, and so forth) are given by the Audit Log Expiration Time Unit parameter.

Default value

1

Valid range

-1 | 1 to 65535

A value of -1 means that an audit log will never be deleted due to its age.

slapd.conf Syntax

```
auditlog-logexpirationtime <integer>
```

Example

```
auditlog-logexpirationtime 1
```

Audit Log Expiration Time Unit

Specifies the units for Audit Log Expiration Time.

Default value

week

Valid range

month | week | day | hour | minute

slapd.conf Syntax

```
auditlog-logexpirationtimeunit <keyword>
```

Example

```
auditlog-logexpirationtimeunit day
```

Audit Log Maximum Disk Space

Specifies the maximum amount of disk space in megabytes that the audit logs are allowed to consume. If this value is exceeded, the oldest audit log is deleted.

When setting a maximum disk space, consider the total number of log files that can be created due to log file rotation. Also remember that there are 3 different log files (access log, audit log, and error log) maintained by the directory server, each of which will consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the audit log.

Default value

500

Valid range

-1 | 1 to 65535

A value of -1 means that the disk space allowed to the audit log is unlimited in size.

slapd.conf Syntax

```
auditlog-maxlogdiskspace <integer>
```

Example

```
auditlog-maxlogdiskspace 500
```

Audit Log Maximum Log Size

Specifies the maximum audit log size in megabytes. When this value is reached, the audit log is rotated. That is, the server starts writing log information to a new log file. If you set “Audit Log Maximum Number of Log Files” to 1, the server ignores this parameter.

Default value

100

Valid range

-1 | 1 to 65535

A value of -1 means the log file is unlimited in size. When setting a maximum log size, consider the total number of log files that can be created due to log file rotation. Also remember that there are 3 different log files (access log, audit log, and error log) maintained by the directory server, each of which will consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the audit log.

slapd.conf Syntax

```
auditlog-maxlogsize <integer>
```

Example

```
auditlog-maxlogsize 100
```

Audit Log Maximum Number of Log Files

Specifies the total number of audit logs that can be contained in the directory where the audit log is stored. If you are using log file rotation, then each time the audit log is rotated, a new log file is created. When the number of files contained in the audit log directory exceeds the value stored on this parameter, then the oldest version of the log file is deleted. The default is 1 log. If you accept this default, the server will not rotate the log and it will grow indefinitely.

Default value

1

Valid range

1 to 65535

slapd.conf Syntax

```
auditlog-maxNumOfLogsPerDir <integer>
```

Example

```
auditlog-maxNumOfLogsPerDir 10
```

Audit Log Minimum Free Disk Space

Specifies the minimum allowed free disk space in megabytes. When the amount of free disk space falls below the value specified on this parameter, the oldest audit log is deleted until enough disk space is freed to satisfy this parameter.

Default value

5

Valid range

1 to 65535

slapd.conf Syntax

```
auditlog-minfreediskspace <integer>
```

Example

```
auditlog-minfreediskspace 5
```

Audit Log Rotation Time

Specifies the time between audit log file rotations. The audit log will be rotated when this time interval is up, regardless of the current size of the audit log. This parameter supplies only the number of units. The units (day, week, month, and so forth) are given by the Audit Log Rotation Time Unit parameter. If you set "Audit Log Maximum Number of Log Files" to 1, the server ignores this parameter.

Default value

1

Valid range

-1 | 1 to 65535

A value of -1 means that the time between audit log file rotation is unlimited.

slapd.conf Syntax

```
auditlog-logrotationtime <integer>
```

Example

```
auditlog-logrotationtime 100
```

Audit Log Rotation Time Unit

Specifies the units for Audit Log Rotation Time.

Default value

day

Valid range

month | week | day | hour | minute

slapd.conf Syntax

```
auditlog-logrotationtimeunit <keyword>
```

Example

```
auditlog-logrotationtimeunit day
```

Certificate and Key Directory

Specifies the location of the SSL directory. This directory contains Secure Socket Layer-related files. This parameter is configurable only from `slapd.conf`; it is not configurable from the server console.

Default value

<NSHOME>/slapd-<serverID>/ssl

Valid range

Currently this directory must be set to the default.

slapd.conf Syntax

```
security-path <string>
```

Example

```
security-path /usr/ns-home/slapd-directory/ssl
```

Changelog DB Directory

Specifies the name of the directory in which the change log database is stored. Netscape recommends that this database be stored in:

```
<NSHOME>/slapd-<serverID>/changelogdb
```

The change log is used to record modifications made to a supplier server's database. When the supplier server's directory is modified, and entry is written to the change log that contains:

- A number that uniquely identifies the modification. This number is sequential with respect to other entries in the change log.
- The modification action; that is, exactly how the directory was modified.

When the supplier server updates a consumer server, the supplier uses the change log information to determine if any modifications have occurred that need to be propagated to the consumer server. If so, the supplier server modifies the consumer server based on the modification(s) recorded in the change log.

This parameter must be set to a valid directory name before replication can occur. For more information on replication, refer to Chapter 13, "Managing Replication."

Default value

null string

Valid range

Any valid file name

slapd.conf Syntax

```
changelogdir <directory>
```

Example

```
changelogdir /usr/ns-home/slapd-local/changelogdb
```

Changelog Suffix

Specifies the suffix used for the change log directory. For information on the change log directory, see "Changelog DB Directory".

Default value

null string

Valid range

Any valid string

slapd.conf Syntax

```
changelogsuffix <suffix>
```

Example

```
changelogsuffix cn=changelog
```

Check Password Syntax

Indicates whether the password syntax will be checked before the password is saved. The password syntax checking mechanism checks that the password meets or exceeds the password minimum length requirement and that the string does not contain any "trivial" words, such as the user's name or user ID or any attribute value stored in the user's directory entry.

For more information on password policies, see Chapter 6, "Managing Password and Account Lockout Policies".

Default Value

Off

Valid Range

On | Off

slapd.conf Syntax

```
pw_syntax <Boolean>
```

Example

```
pw_syntax off
```

Enable Access Control

Turns access control on and off. If this parameter is set to off, then any valid bind attempt (including anonymous binds) results in full access to all information stored in the directory server.

Default Value

on

Valid Range

on | off

slapd.conf Syntax

```
accesscontrol <Boolean>
```

Example

```
accesscontrol off
```

Enable Online Consumer Creation

Indicates whether a server will automatically use online consumer (replica) creation in the event that an inconsistency is detected between the databases on the supplier and the consumer servers. If this parameter is missing from slapd.conf, or if this parameter is set to anything other than on, then online consumer creation is turned off.

Online consumer creation applies to both supplier-initiated replication and consumer-initiated replication. If supplier-initiated replication is used, then online consumer creation is either turned on or off for all consumer servers. Similarly, if consumer-initiated replication is used then online consumer creation is either turned on or off for all supplier servers.

Caution should be used before turning this feature on. For more information, see “Initializing Consumers” on page 344.

This parameter is not available from the server console.

Default Value

off

Valid Range

on | off

slapd.conf Syntax

```
orcauto <Boolean>
```

Example

```
ntsynch-port on
```

Enable Superior Object Class Enquoting

Controls whether quoting in the `objectclasses` attributes contained in the `cn=schema` entry will conform to the quoting specified by internet draft RFC 2252. By default, the Directory Server places single quotes around the superior object class identified on the `objectclasses` attributes contained in `cn=schema`. RFC 2252 indicates that this value should not be quoted.

That is, the Directory Server publishes `objectclasses` attributes in the `cn=schema` entry as follows:

```
objectclasses: ( 2.5.6.6 NAME 'person' DESC 'Standard ObjectClass' SUP
'top' MUST ( objectclass $ sn $ cn ) MAY ( aci $ description $ seealso $
telephonenumber $ userpassword ) )
```

However, RFC 2252 indicates that this attribute should be published as follows:

```
objectclasses: ( 2.5.6.6 NAME 'person' DESC 'Standard ObjectClass' SUP
top MUST ( objectclass $ sn $ cn ) MAY ( aci $ description $ seealso $
telephonenumber $ userpassword ) )
```

Notice the lack of single quotes around the word `top`.

Turning this parameter off causes the Directory Server to conform to RFC 2252, but doing so may interfere with an LDAP client's ability to modify schema. Specifically, any client written using the Netscape Java LDAP SDK will no longer be able to correctly read and modify schema. This includes the 4.x version of the Netscape Console.

In addition, any LDAP clients that use the C LDAP SDK may no longer be able to correctly manage schema (unlike the Java LDAP SDK, the C SDK does not include standard routines for schema management, so the effects of this parameter on C SDK-based clients will vary depending on the actual implementation).

Default value

on

Valid range

on | off

slapd.conf Syntax

```
enquote_sup_oc <Boolean>
```

Example

```
enquote_sup_oc on
```

Encrypted Port Number

TCP/IP port number used for SSL communications. This selected port must be unique on the host system; make sure no other application is attempting to use the same port number. For UNIX systems, specifying a port number of less than 1024 requires that the administration server run as root, because it must start the directory server with root privileges.

Default value

636

Valid range

1 to 65535

slapd.conf Syntax

```
secure-port <integer>
```

Example

```
secure-port 636
```

Encryption Alias

The encryption alias you want to use for this server's certificate. You create the encryption alias when you create your server's certificate database. For more information on creating certificate databases, see the "Enabling SSL Encryption" section in *Managing Servers with Netscape Console*.

Default value

none

Valid range

Any valid string

slapd.conf Syntax

```
encryption-alias <string>
```

Example

```
encryption-alias secure-LDAP
```

Encryption Ciphers

Specifies the type of encryption the directory server will use when using SSL communications. For more information on the ciphers supported by the directory server, refer to Chapter 11, "Managing SSL."

Default value

N/A

Valid range

For domestic versions, any combination of the following:

- RC4-40 MD5
- RC2-40 MD5
- RC4-128 MD5
- FIPS DES-56 SHA
- Triple DES-168 SHA
- FIPS Triple DES-168 SHA
- DES-56 SHA
- clear MD5

For export versions, any combination of the following:

- RC4-40 MD5
- RC2-40 MD5
- clear MD5

slapd.conf Syntax

SSL3ciphers <cipher>[,<cipher>, <cipher>, . . .]

where <cipher> is any of the following:

- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_FIPS_DES_56_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_FIPS_3DES_168_SHA

- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_NULL_MD5

Export versions can use only the following ciphers:

- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- SSL_RSA_WITH_NULL_MD5

White spaces are not allowed in the list of ciphers.

Example

```
SSL3cipher SSL_RSA_EXPORT_WITH_RC4_40_MD5,SSL_RSA_
EXPORT_WITH_RC2_CBC_40_MD5
```

Error Log

Specifies the pathname and filename of the log used to record error messages generated by the directory server. These messages can describe error conditions, but more often they will contain informative conditions such as:

- server startup and shutdown times
- port number the server uses

This log will contain differing amounts of information depending on the current setting of the Log Level parameter. See “Log Level” on page 444 for more information on the Log Level parameter.

Default value

<NSHOME>/slapd-<serverID>/logs/error

Valid range

Any valid filename

slapd.conf Syntax

```
errorlog <filename>
```

Example

```
errorlog /usr/ns-home/slapd-<serverId>/logs/error
```

Error Log Enable Logging

Turns error logging on and off.

Default value

on

Valid range

on | off

slapd.conf Syntax

```
errorlog-logging-enabled <Boolean>
```

Example

```
errorlog-logging-enabled on
```

Error Log Expiration Time

Specifies the maximum age that a log file is allowed to be before it is deleted. This parameter supplies only the number of units. The units (day, week, month, and so forth) are given by the Error Log Expiration Time Unit parameter.

Default value

1

Valid range

-1 | 1 to 65535

A value of -1 means that an error log will never be deleted due to its age.

slapd.conf Syntax

```
errorlog-logexpirationtime <integer>
```

Example

```
errorlog-logexpirationtime 1
```

Error Log Expiration Time Unit

Specifies the units for Error Log Expiration Time.

Default value

week

Valid range

month | week | day | hour | minute

slapd.conf Syntax

```
errorlog-logexpirationtimeunit <keyword>
```

Example

```
errorlog-logexpirationtimeunit day
```

Error Log Maximum Disk Space

Specifies the maximum amount of disk space in megabytes that the error logs are allowed to consume. If this value is exceeded, the oldest error log is deleted.

When setting a maximum disk space, consider the total number of log files that can be created due to log file rotation. Also remember that there are 3 different log files (access log, audit log, and error log) maintained by the directory server, each of which will consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the error log.

Default value

500

Valid range

-1 | 1 to 65535

A value of -1 means that the disk space allowed to the error log is unlimited in size.

slapd.conf Syntax

```
errorlog-maxlogdiskspace <integer>
```

Example

```
errorlog-maxlogdiskspace 500
```

Error Log Maximum Log Size

Specifies the maximum error log size in megabytes. When this value is reached, the error log is rotated. That is, the server starts writing log information to a new log file. If you set “Error Log Maximum Number of Log Files” to 1, the server ignores this parameter.

Default value

100

Valid range

-1 | 1 to 65535

A value of -1 means the log file is unlimited in size. When setting a maximum log size, consider the total number of log files that can be created due to log file rotation. Also remember that there are 3 different log files (access log, audit log, and error log) maintained by the directory server, each of which will consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the error log.

slapd.conf Syntax

```
errorlog-maxlogsize <integer>
```

Example

```
errorlog-maxlogsize 100
```

Error Log Maximum Number of Log Files

Specifies the total number of error logs that can be contained in the directory where the error log is stored. If you are using log file rotation, then each time the error log is rotated, a new log file is created. When the number of files contained in the error log directory exceeds the value stored on this parameter, then the oldest version of the log file is deleted. The default is 1 log. If you accept this default, the server will not rotate the log and it will grow indefinitely.

Default value

1

Valid range

1 to 65535

slapd.conf Syntax

```
errorlog-maxNumOfLogsPerDir <integer>
```

Example

```
errorlog-maxNumOfLogsPerDir 10
```

Error Log Minimum Free Disk Space

Specifies the minimum allowed free disk space in megabytes. When the amount of free disk space falls below the value specified on this parameter, the oldest error log is deleted until enough disk space is freed to satisfy this parameter.

Default value

5

Valid range

1 to 65535

slapd.conf Syntax

```
errorlog-minfreediskspace <integer>
```

Example

```
errorlog-minfreediskspace 5
```

Error Log Rotation Time

Specifies the time between error log file rotations. The error log will be rotated when this time interval is up, regardless of the current size of the error log. This parameter supplies only the number of units. The units (day, week, month, and so forth) are given by the Error Log Rotation Time Unit parameter. If you set "Error Log Maximum Number of Log Files" to 1, the server ignores this parameter.

Default value

1

Valid range

-1 | 1 to 65535

A value of -1 means that the time between error log file rotation is unlimited.

slapd.conf Syntax

```
errorlog-logrotationtime <integer>
```

Example

```
errorlog-logrotationtime 100
```

Error Log Rotation Time Unit

Specifies the units for Error Log Rotation Time.

Default value

day

Valid range

month | week | day | hour | minute

slapd.conf Syntax

```
errorlog-logrotationtimeunit <keyword>
```

Example

```
errorlog-logrotationtimeunit day
```

Idle Timeout

Specifies the amount of time in seconds after which an idle LDAP client connection is closed by the server. A value of 0 indicates that the server will never close idle connections.

Default Value

0

Valid Range

0 - maximum integer

slapd.conf Syntax

```
idletimeout <integer>
```

Example

```
idletimeout 0
```

Instance Directory

Specifies the full path to the directory where this server instance is installed.

Default Value

<NSHOME>/slapd-<server ID>

Valid Range

Any valid file path.

slapd.conf Syntax

```
instancedir “<file path>”
```

Example

```
instancedir “/user/netscape/slapd-phonebook”
```

IO Block Time Out

Specifies the amount of time in milliseconds after which the connection to a stalled LDAP client is closed. An LDAP client is considered to be stalled when it has not made any I/O progress for read or write operations.

Default Value

1800000 (30 minutes)

Valid Range

0 - maximum integer

slapd.conf Syntax

```
ioblocktimeout <integer>
```

Example

```
ioblocktimeout 1800000
```

Listen to IP Address

Used only on multihomed machines. The directory server will only respond to requests sent to the interface that correspond to the IP address provided on this parameter.

Default Value

N/A

Valid Range

Any IP address configured for the local host.

slapd.conf Syntax

```
listenhost <IP address>
```

Example

```
listenhost 111.11.111.1
```

Local User

Unix installations only. Specifies the user that the directory server runs as. The group that the user runs as is derived from this parameter by examining the groups that the user is a member of.

Default Value

N/A

Valid Range

Any valid user on the local Unix machine.

slapd.conf Syntax

```
localuser <string>
```

Example

```
localuser nobody
```

Lockout Duration

Indicates the amount of time in seconds that users will be locked out of the directory after an account lockout. The account lockout feature protects against hackers who try to break into the directory by repeatedly trying to guess a user's password. You enable and disable the account lockout feature using the Account Lockout parameter. Instead of locking out users for a specified amount of time, you can choose to lock users out until the administrator resets the password using the Unlock Account parameter.

For more information on password policies, see Chapter 6, "Managing Password and Account Lockout Policies".

Default Value

3600 seconds

Valid Range

1 to maximum integer

slapd.conf Syntax

```
pw_lockoutduration <integer>
```

Example

```
pw_lockoutduration 3600
```

Log Buffering

When this option is set to “off”, the server writes all access log entries directly to disk.

Default value

None

Valid range

On | Off

slapd.conf Syntax

```
logbuffering <Boolean>
```

Example

```
logbuffering off
```

Log Level

Specifies the level of logging to be used by the directory server. The log level is additive; that is, specifying a value of 3 causes both levels 1 and 2 to be performed.

To turn off logging, remove the loglevel parameter from slapd.conf and restart the directory server.

Default value

Logging is turned off (the `loglevel` parameter is not included in slapd.conf).

Valid range

1—Trace function calls. Creates an entry when the server enters and exits a function.

2—Debug Packet handling

4—Heavy trace output debugging

8—Connection management

16—Print out packets sent/received

32—Search filter processing

64—Config file processing

128—Access control list processing

1024—Log communications with shell backends

2048—Log entry parsing debugging

4096—Housekeeping thread debugging

8192—Replication debugging

16386—Generic debugging; a catch all for the debugging that does not fit in any of the other categories.

32768—Database cache debugging.

65536— Server plug-in debugging; writes an entry to the log file when a server plug-in calls `slapi_log_error`. For information on server plugins, see the *Netscape Directory Server Programmer's Guide*.

slapd.conf Syntax

```
loglevel <integer>
```

Example

```
loglevel 8192
```

Max Changelog Age

Specifies the maximum age of any entry in the change log. The change log contains a record for each directory modification and is used when synchronizing consumer servers. Each record contains a timestamp. Any record with a timestamp that is older than the value specified in this parameter will be removed. If this parameter is absent, there is no age limit on change log records. For information on the change log, see ""Changelog DB Directory" on page 426."

Default

none

Valid range

0 to maximum integer

slapd.conf Syntax

```
changelogmaxage <integer><Age ID>
```

where Age ID is “s” for seconds, “m” for minutes, “h” for hours, “d” for days, or “w” for weeks.

Example

```
changelogmaxage 30d
```

Max Changelog Records

Specifies the maximum number of records the change log may contain. If this parameter is absent, there is no maximum number of records the change log can contain. For information on the change log, see ““Changelog DB Directory” on page 426.”

Default value

none

Valid range

0 to 65535

slapd.conf Syntax

```
changelogmaxentries <integer>
```

Example

```
changelogmaxentries 5000
```

Maximum File Descriptors

Not applicable to directory installations on NT and AIX.

This parameter sets the maximum number of file descriptors that the directory server will try to use. A file descriptor is used whenever a client connects to the server, and for some server activities such as index maintenance.

The number that you specify here should not be greater than the total number of file descriptors that your operating system allows the `ns-slapd` process to use. This number will differ depending on your operating system. Some operating systems allow you to configure the number of file descriptors available to a process. See your operating system documentation for details on file descriptor limits and configuration.

You should consider increasing the value on this parameter if the directory server is refusing connections because it is out of file descriptors. When this condition occurs, the following message is written to the directory server's error log file:

```
Not listening for new connections -- too many fds open
```

Default Value

1024

Valid Range

1 to 65535

slapd.conf Syntax

```
maxdescriptors <integer>
```

Example

```
maxdescriptors 1024
```

Maximum Message Size

Defines the maximum size in bytes allowed by an incoming message. This limits the size of write operations to the directory server. Limiting the size of write operations prevents some kinds of denial of service attacks.

The write applies to the total size of the add or modify. For example, if a new entry contains five attributes and the sum of those five attributes exceeds this limit, then the add is denied.

Do not change this parameter value unless told to do so by Netscape support personnel.

A value of 0 indicates that the default value should be used.

Default Value

2097152

Valid Range

0 - 2^{32}

slapd.conf Syntax

```
maxbersize <bytes>
```

Example

```
maxbersize 2097152
```

Maximum Password Failures

Indicates the number of failed bind attempts after which a user will be locked out of the directory. By default, account lockout is disabled. You can enable account lockout by modifying the Account Lockout parameter.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

3 bind failures

Valid Range

1 to maximum integer

slapd.conf Syntax

```
pw_maxfailure <integer>
```

Example

```
pw_maxfailure 3
```

Maximum Threads Per Connection

Defines the maximum number of threads that a connection should use. For normal operations where a client binds and only performs one or two operations before unbinding, you should use the default value. For situations where a client binds and does many operations, you should increase this value to allow each connection enough resources to perform all the operations.

A value of 0 turns off maxthreadsperconn and causes the server to allow each connection to obtain as many threads as the connection requires, up to the value set by Thread Number.

This parameter is not available from the server console.

Default value

5

Valid range

0 to threadnumber

slapd.conf Syntax

```
maxthreadsperconn <number of threads>
```

Example

```
maxthreadsperconn 5
```

nagle

Reserved for future use. Do not change or remove. Doing so can have unpredictable results.

NLS

Used to define the directory where the internationalization files are kept.

This parameter is not available from the server console.

Default value

<NSHOME>/nls

Valid range

N/A

slapd.conf Syntax

```
NLS “<directory>”
```

Example

```
NLS “/usr/ns-home/nls”
```

NT Synchronization Service Enabled

Indicates whether the NT Synchronization Service plug-ins are used by the directory server. These plug-ins cause the directory server to validate all NT Directory Data with the appropriate NT primary domain controller. These plug-ins also transfer NT user and group changes to the synchronization service for inclusion on NT user and group accounts.

For more information on the NT synchronization service, see Chapter 15, “NT Directory Synchronization.”

Default Value

No

Valid Range

Yes | No

slapd.conf Syntax

```
ntsynch on|off
```

Example

```
ntsynch on
```

NT Synchronization Service Port Number

Specifies the port number that directory server will use for non-LDAP communications with the NT Synchronization Service. This port is used to validate directory changes with the NT domain, and to transfer directory changes to NT. For UNIX systems, specifying a port number of less than 1024 requires that administration server run as root, because it must start the directory server with root privileges.

For more information on the NT synchronization service, see Chapter 15, “NT Directory Synchronization.”

Default Value

5005

Valid Range

1 to 65535

slapd.conf Syntax

```
ntsynch-port <integer>
```

Example

```
ntsynch-port 5005
```

NT Synchronization Service Use SSL

Indicates whether the directory server will use SSL when communicating with the NT synchronization service. This parameter applies to both LDAP and non-LDAP communications.

If you change this parameter, make sure you make the corresponding change in the NT Synchronization Service Configuration tool. Also, if this parameter is set to Off and you turn it on, make sure that your directory server is configured for use with SSL.

Default Value

on

Valid Range

on | off

slapd.conf Syntax

```
ntsynchusessl <Boolean>
```

Example

```
ntsynchusessl on
```

Number of Passwords to Remember

Indicates the number of passwords the directory server stores in history. Passwords that are stored in history cannot be reused by users. By default, the password history feature is disabled. That is, the directory server does not store any old passwords and so users can reuse passwords. You can enable password history by using the Password History parameter.

To prevent users from rapidly cycling through the number of passwords that you are tracking, use the Password Minimum Age parameter.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

6 passwords

Valid Range

2 to 24 passwords

slapd.conf Syntax

```
pw_inhistory <integer>
```

Example

```
pw_inhistory 6
```

Object Class

Used to define the schema rules for the specified object class. This parameter is intended to allow the extension of the standard schema when schema checking is turned on.

This parameter is not available from the server console.

slapd.conf Syntax

```
objectClass <name>
  oid <oid number>
```

```
  superior <superior object class>
  requires <list of attributes>
  allows <list of attributes>
```

Example

```
objectClass person
  requires
    objectClass,
    sn,
    cn

  allows
    description,
    seeAlso,
    telephoneNumber,
    userPassword,
    subtreeACI
```

Password Change

Indicates whether users may change their passwords.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

on

Valid Range

on | off

slapd.conf Syntax

```
pw_change <Boolean>
```

Example

```
pw_change on
```

Password Expiration

Indicates whether user passwords will expire after a given number of seconds. By default, user passwords do not expire. Once password expiration is enabled, you can set the number of seconds after which the password will expire using the Password Maximum Age parameter.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

`Off`

Valid Range

`On|Off`

slapd.conf Syntax

`pw_exp <Boolean>`

Example

`pw_exp on`

Password History

Enables password history. Password history refers to whether users are allowed to reuse passwords. By default, password history is disabled and users can reuse passwords. If you set this parameter to be on, the directory stores a given number of old passwords and prevents users from reusing any of the stored passwords. You set the number of old passwords the directory server stores using the Number of Passwords to Remember parameter.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

Off

Valid Range

On | Off

slapd.conf Syntax

```
pw_history <Boolean>
```

Example

```
pw_history on
```

Password Maximum Age

Indicates the number of seconds after which user passwords will expire. To use this parameter, you must enable password expiration using the Password Expiration parameter.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

8640000 seconds (100 days)

Valid Range

1 to maximum integer

slapd.conf Syntax

```
pw_maxage <integer>
```

Example

```
pw_maxage 100
```

Password Minimum Age

Indicates the number of seconds that must pass before a user can change their password. Use this parameter in conjunction with the Number of Passwords to Remember parameter to prevent users from quickly cycling through passwords so that they can use their old password again. A value of zero (0) indicates that the user can change the password immediately.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

0

Valid Range

0 to 2147472000 seconds (24,855 days)

slapd.conf Syntax

```
pw_minage <integer>
```

Example

```
pw_minage 86400
```

Password Minimum Length

Specifies the minimum number of characters that must be used in directory server passwords. In general, shorter passwords are easier to crack, so you are recommended to set a password length of at least 6 or 7 characters. This is long enough to be difficult to crack, but short enough that users can remember the password without writing it down.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

6 characters

Valid Range

2 to 512 characters

slapd.conf Syntax

```
pw_minlength <integer>
```

Example

```
pw_minlength 6
```

Password Must Change

Indicates whether users must change their passwords when they first bind to the directory server.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

off

Valid Range

on | off

slapd.conf Syntax

```
pw_must_change <Boolean>
```

Example

```
pw_must_change off
```

Password Storage Scheme

Specifies the type of encryption used to store directory server passwords. A null string entered in the form of two double quotation marks ("") indicates that passwords are to be stored in plain text.

The following encryption types are available:

- sha is the Secure Hash Algorithm. It is the most secure of the choices and is the one Netscape recommends.
- crypt is the UNIX crypt algorithm. It is provided for compatibility with UNIX passwords.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default value

sha

Valid range

crypt | sha | ""

slapd.conf Syntax

```
pw_storagescheme <string>
```

Example

```
pw_storagescheme sha
```

Port Number

TCP/IP port number used for non-SSL communications. This selected port must be unique on the host system; make sure no other application is attempting to use the same port number. On UNIX systems, specifying a port number of less than 1024 requires that the administration server run as root, because it must start the directory server with root privileges.

If you are changing the port number for a configuration directory, you must also update the corresponding SIE in the configuration directory.

Default value

389

Valid range

1 to 65535

slapd.conf Syntax

```
port <integer>
```

Example

```
port 389
```

Referral

Specifies the default LDAP URL to pass back when the receives a request for an entry that is not a member of the local tree, that is, an entry whose suffix does not match the value specified on any of the Suffix parameters. For example, suppose the local database contains only entries:

```
ou=People, o=Airius.com
```

but the request is for this entry:

```
ou=Groups, o=Airius.com
```

In this case, the referral would be passed back to the client in an attempt to allow the LDAP client to locate a database that contains the requested entry.

If a smart referral can be found to return as a result of the request, the server will return that referral instead of the value specified on this parameter.

Only one referral is allowed per directory server instance.

For more information on managing referrals, see Chapter 14, “Managing Referrals.”

Default value

null string

Valid range

Any LDAP URL of the form:

```
ldap://<server location>
```

If you want to use SSL communications, the Referral parameter should be of the form:

```
ldaps://<server location>
```

slapd.conf Syntax

```
referral <url>
```

Example

```
referral ldap://ldap.aceindustry.com
```

Reserved File Descriptors

Not applicable to directory installations on NT and AIX.

This parameter sets the number of file descriptors that the directory server reserves for managing non-client connections, such as index management and managing replication. The number of file descriptors that you reserve for this purpose subtracts from the total number of file descriptors available for servicing LDAP client connections (see “Maximum File Descriptors” on page 447).

Most installations of the directory server should never need to change this parameter. However, consider increasing the value on this parameter if all of the following are true:

- The server is replicating to a large number of consumer servers (more than 10) and/or the server is maintaining a large number of index files (more than 30).
- The server is servicing a large number of LDAP connections
- You are seeing error messages reporting that the server is unable to open file descriptors (the actual error message will differ depending on the operation that the server is attempting to perform), but these error messages are NOT related to managing client LDAP connections.

Increasing the value on this parameter may result in more LDAP clients being unable to access your directory. Therefore, when you increase the value on this parameter, you should also increase the value on the Maximum File Descriptors parameter. You may not be able to increase the Maximum File Descriptors

value if your server is already using the maximum number of file descriptors that your operating system allows a process to use (see your operating system documentation for details). If this is the case, then reduce the load on your server by causing LDAP clients to search alternative directory replicas.

Default Value

64

Valid Range

1 to 65535

slapd.conf Syntax

```
reservedescriptors <integer>
```

Example

```
reservedescriptors 64
```

Reset Password Failure Count After

Indicates the amount of time in seconds after which the password failure counter will be reset. Each time an invalid password is sent from the user's account, the password failure counter is incremented. If the account lockout feature is enabled, users will be locked out of the directory when the counter reaches the number of failures specified by the Maximum Password Failures parameter within 600 seconds by default. After 600 seconds, the failure counter will be reset to zero (0). You enable or disable the account lockout feature using the Account Lockout parameter.

For more information on password policies, see Chapter 6, "Managing Password and Account Lockout Policies".

Default Value

600 seconds

Valid Range

1 to maximum integer

slapd.conf Syntax

```
pw_resetfailurecount <integer>
```

Example

```
pw_resetfailurecount 600
```

result_tweak

Reserved for future use. Do not change or remove. Doing so can have unpredictable results.

Root DN

Specifies the distinguished name of an entry that is not subject to access control or administrative limit restrictions for operations on the database. Size Limit, Time Limit, and Schema Checking also do not apply to this DN.

For information on changing the Root DN, see “Managing the Root DN” on page 288.

Valid range

Any valid distinguished name.

slapd.conf Syntax

rootdn <“string”>

Example

```
rootdn “cn=Directory Manager, o=airius.com”
```

Root Password

When viewed from the server console, this parameter shows the value: “Not Displayed.” When viewed from the `slapd.conf` file, this parameter shows the encryption method followed by the encrypted string.

Warning

If you configure a root DN at server installation time, you must also provide a root password. However, it is possible for the root password to be deleted from `slapd.conf` by direct editing of the file. In this situation, the root DN can only obtain the same access to your directory as you allow for anonymous access. Always make sure that a root password is defined in `slapd.conf` when a root DN is configured for your database.

Valid range

Any valid password. Possible encryption methods are described in “Password Storage Scheme” on page 458.

slapd.conf Syntax

```
rootpw <{encryption method}encrypted password>
```

Example

```
rootpw {crypt}9Eko69APCJfF
```

Root Password Storage Scheme

Available only from the server console. This parameter indicates the encryption method used for the root password.

Default value

Clear text

Valid range

Any encryption method as described in “Password Storage Scheme” on page 458.

slapd.conf Syntax

```
rootpw {encryption method}encrypted password
```

Example

```
rootpw {crypt}9Eko69APCJfF
```

Schema Checking

Specifies whether the database schema will be enforced during entry insertion or modification. The database schema defines the type of information allowed in the database. You can extend the default schema using the `objectclass` and attribute parameters. For information on how to extend your schema using the Directory Server Console, see Chapter 3, "Extending the Directory Schema."

Note Schema checking works by default when database modifications are made using an LDAP client, such as `ldapmodify`, the directory server gateway, or when importing a database from LDIF using `ldif2db`. If you turn schema checking off, you will manually have to verify that your entries conform to the schema. Make sure that the attributes and object classes you create in your LDIF statements are both spelled correctly and are identified in `slapd.conf`, `slapd.at.conf`, or `slapd.oc.conf`, or a custom schema file that you are including into `slapd.conf`.

Default value

on

Valid range

on | off

slapd.conf Syntax

```
schemacheck <Boolean>
```

Example

```
schemacheck on
```

Security

Specifies whether the directory server is to accept SSL communications on its encrypted port.

Default value

off

Valid range

on | off

slapd.conf Syntax

```
security <Boolean>
```

Example

```
security off
```

Send Warning

Indicates the number seconds before a user's password is due to expire that the user will be sent a warning message. Depending on the LDAP client, the user may also be prompted to change their password at the time the warning is sent.

For more information on password policies, see Chapter 6, "Managing Password and Account Lockout Policies".

Default Value

86400 seconds (1 day)

Valid Range

1 to maximum integer

slapd.conf Syntax

```
pw_warning <integer>
```

Example

```
pw_warning 86400
```

Size Limit

Specifies the maximum number of entries to return from a search operation. If this limit is reached, `ns-slapd` returns any entries it has located that match the search request, as well as an exceeded size limit error.

A null string on this parameter causes no limit to be used; `ns-slapd` will return every matching entry to the client regardless of the number found. To set this no limit value from within `slapd.conf`, specify a negative value on the parameter. A value of zero (0) causes no entries to be returned for searches.

Default value

2000

Valid range

-1 to 65535

A value of -1 on this parameter in `slapd.conf` is the same as leaving the parameter blank in the server console; it causes no limit to be used. However, you cannot specify a negative integer for this field in the server console; nor can you specify a null value in `slapd.conf`.

slapd.conf Syntax

```
sizelimit <integer>
```

Example

```
sizelimit 2000
```

Supplier DN

Specifies the distinguished name that supplier servers use to update your server with replicated data. For more information on replication, refer to Chapter 13, “Managing Replication”.

Default value

null string

Valid range

Any valid distinguished name representing an entry in the local directory tree.

slapd.conf Syntax

```
updatedn <“DN”>
```

Example

```
updatedn “cn=Replication Admin, o=Airius.com”
```

Supplier Password

The password the consumer server expects the supplier server to use when binding. The supplier password is only required if the consumer server is not configured to accept certificate-based authentication.

Default value

null string

Valid range

Any valid password of 8 or more characters. Possible Encryption methods are described in “Password Storage Scheme” on page 458.

slapd.conf Syntax

```
updatepw <{encryption method} encrypted password>
```

Example

```
updatepw {crypt} 9EKo74BXRKnL
```

Supplier SSL Clients

The subject name(s) of the certificate(s) that correspond to the supplier DN defined for the consumer server. If a client sends a certificate with a subject name that matches any of the subject names configured for this parameter, the client is automatically authenticated as the supplier DN. This parameter is only

used when the consumer server is configured to accept certificate-based authentication and when a supplier DN is defined. The value of this parameter must match the certificate subject name exactly; differences in case or whitespace are significant.

Default value

null string

Valid range

Any valid certificate subject DN.

slapd.conf Syntax

```
updateSSLclient <certificate subject DN>
```

Example

```
updateSSLclient “cn=master.airius.com, o=airius.com”
```

Thread Number

Defines the number of operation threads that the directory server will create during start up.

This parameter is not available from the server console.

Default value

20

Valid range

1 to the number of threads supported by your system

slapd.conf Syntax

```
threadnumber <number threads>
```

Example

```
threadnumber 20
```

Time Limit

Specifies the maximum number of seconds allocated for a search request. If this limit is reached, the directory server returns any entries it has located that match the search request, as well as an exceeded time limit error.

A null string on this parameter causes no limit to be used; the directory server will wait indefinitely for the search to complete. To set this no limit value from within `slapd.conf`, specify a negative value on the parameter. A value of zero (0) causes no time to be allowed for searches.

Default value

3600

Valid range

-1 to 65535

A value of -1 on this parameter in `slapd.conf` is the same as leaving the parameter blank in the server console; it causes no limit to be used. However, you cannot specify a negative integer for this field in the server console; nor can you specify a null value in `slapd.conf`.

slapd.conf Syntax

```
timelimit <integer>
```

Example

```
timelimit 3600
```

Track Modification Time

Specifies whether the directory server maintains the modification attributes for directory server entries. These attributes include:

- `modifiersname`—The distinguished name of the person who last modified the entry.
- `modifytimestamp`—The timestamp for when the entry was last modified in GMT format.

- `creatorsname`—The distinguished name of the person who initially created the entry.
- `createtimestamp`—The timestamp for when the entry was created in GMT format.

If you are using your directory server with the NT user synchronization, then this parameter must be turned on.

Default value

on

Valid range

on | off

slapd.conf Syntax

```
lastmod <Boolean>
```

Example

```
lastmod off
```

Unlock Account

Indicates whether users will be locked out of the directory for a specified amount of time or until the administrator resets the password after an account lockout. The account lockout feature protects against hackers who try to break into the directory by repeatedly trying to guess a user's password. You enable and disable the account lockout feature using the Account Lockout parameter. Instead of locking users out forever, you can choose to lock users out for a specified amount of time using the Lockout Duration parameter.

For more information on password policies, see Chapter 6, “Managing Password and Account Lockout Policies”.

Default Value

On

Valid Range

On | Off

slapd.conf Syntax

```
pw_unlock <Boolean>
```

Example

```
pw_unlock off
```

User-Defined Attributes File

Provides the full path name to the locally-defined attributes. Use this file when extending the attributes in your schema.

Default Value

<NSHOME>/slapd-<server ID>/config/slapd.user_at.conf

Valid Range

Any valid file path.

slapd.conf Syntax

```
userat "<file path>"
```

Example

```
userat
"/user/netscape/slapd-phonebook/config/slapd.user_at.conf
"
```

User-Defined Object Class File

Provides the full path name to the locally-defined attributes. Use this file when extending the object classes in your schema.

Default Value

<NSHOME>/slapd-<server ID>/config/slapd.user_oc.conf

Valid Range

Any valid file path.

slapd.conf Syntax

```
useroc "<file path>"
```

Example

```
userat "/user/netscape/slapd-phonebook/config/slapd.user_at.conf"
```

Database Parameters

Table 17.2 describes the server parameters that apply to the directory server database. These parameters are stored in `slapd.ldbm.conf`.

Table 17.2 Directory server database parameters

Parameter	Description
All IDs Threshold	Specifies the total number of entry IDs that an index key is allowed to manage before the all IDs token is set.
Attribute to be Indexed	String specifying the indexes to maintain for a given attribute.
Database	String marking the beginning of a new database instance definition within `slapd.ldbm.conf`.
Database Checkpoint Interval	The amount of time in seconds after which the directory server sends a checkpoint entry to the database transaction log.
Database Configuration File	Specifies the path to the file containing database parameters.
Database Directory	String specifying the directory that contains the database and associated indexes.
Database Durable Transactions	Indicates whether database transaction log entries are immediately written to the disk.

Table 17.2 Directory server database parameters (Continued)

Parameter	Description
Database Transaction Log Directory	Specifies the path and directory name of the directory containing the database transaction log.
db_home_directory	Solaris-only parameter used for a fix to a Solaris page flushing problem.
Look Through Limit	Integer specifying the maximum number of entries that the directory server will check before returning a resource limit error.
Maximum Cache Size	Integer specifying the size in bytes of the in-memory cache.
Maximum Entries in Cache	Integer specifying the number of entries to be contained in the in-memory cache.
Mode	Integer specifying the file protection used for newly created database index files.
Read-only	Boolean indicating whether the database is in read-only mode.
Suffix	String specifying the distinguished name suffix used for the local database.

All IDs Threshold

Specifies the number of entry IDs that can be maintained for an index key before the server sets the all IDs token. This value should be roughly 5% of the total number of directory entries stored on your server.

For information about all IDs threshold, see “Managing All IDs Threshold” on page 197.

Default value

4000

Valid range

100 to maximum integer

slapd.ldbm.conf Syntax

```
allidsthreshold <integer>
```

Example

```
allidsthreshold 4000
```

Attribute to be Indexed

Specifies the indexes to maintain for the specified attribute(s). If only a list of attributes is provided, all possible indexes are maintained. If a value of default is provided in the place of a list of attributes, all attributes are indexed.

Valid indexes include:

- pres
- eq
- approx
- sub
- none

For a complete description of indexing, refer to Chapter 7, “Managing Indexes.”

Default value

Only default indexing is performed.

Valid range

Any valid attribute and any valid index type. For a list of the commonly used attributes, see the *Netscape Directory Server Schema Reference Guide*.

slapd.ldbm.conf Syntax

```
index [<attribute list>|default] [<list of indexes>]
```

Example

```
index cn
```

```
index sn,uid eq,sub,approx
index default none
```

This example causes all indexes to be maintained for the `cn` attribute; equality, substring, and approximate indexes for the `sn` and `uid` attributes; and no indexes for all other attributes.

Database

Marks the beginning of the database definition in the `slapd.ldbm.conf` file. This parameter is not available from the server console.

Default value

ldbm

Valid range

Currently only `ldbm` is supported.

slapd.ldbm.conf Syntax

```
database ldbm
```

Database Checkpoint Interval

The amount of time in seconds after which the directory server sends a checkpoint entry to the database transaction log. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only. A checkpoint entry indicates which database operations have been physically written to the directory database. The checkpoint entries are used to determine where in the database transaction log to begin recovery after a system failure. The db_checkpoint_interval parameter is absent from `slapd.conf`. To change the checkpoint interval, you add the parameter to `slapd.conf`.

For more information on database transaction logging, see “Managing Database Transaction Logging” on page 93.

Default Value

60 seconds

Valid Range

10 to 300 seconds

Slapd.ldbm.conf Syntax

```
db_checkpoint_interval <integer>
```

Example

```
db_checkpoint_interval 120
```

Database Configuration File

Specifies the path to `slapd.ldbm.conf`, which is a file that contains `slapd.conf` server parameters that can be changed dynamically. Currently only the index parameter is supported in `slapd.ldbm.conf`.

For more information about `slapd.ldbm.conf`, see “Changing Parameter Values Using slapd.ldbm.conf” on page 404.

Default Value

```
<NSHOME>/slapd-<serverID>/config/slapd.ldbm.conf
```

Valid Range

Any valid path and directory name.

Slapd.ldbm.conf Syntax

```
dynamicconf <filename>
```

Example

```
dynamicconf /usr/ns-home/slapd-fire/config/slapd.ldbm.conf
```

Database Directory

Specifies the directory containing the database and associated index files.

Default value

<NSHOME>/slapd-<serverID>/db

Valid range

N/A

slapd.ldbm.conf Syntax

```
directory <string>
```

Example

```
directory /usr/ns-home/slapd-myserver/db
```

Database Durable Transactions

Indicates whether database transaction log entries are immediately written to the disk. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only. With durable transactions enabled, every directory change will always be physically recorded in the log file and therefore be able to be recovered in the event of a system failure. However, the durable transactions feature may also slow the performance of the directory server. When durable transactions is disabled, all transactions are logically written to the database transaction log but may not be physically written to disk immediately. If there was a system failure before a directory change was physically written to disk, that change would not be recoverable. The db_durable_transactions parameter is absent from `slapd.conf`. To disable durable transactions, you add the parameter to `slapd.conf`.

For more information on database transaction logging, see “Managing Database Transaction Logging” on page 93.

Default Value

On

Slapd.ldbm.conf Syntax

db_durable_transactions on | off

Example

```
db_durable_transactions off
```

Database Transaction Log Directory

Specifies the path and directory name of the directory containing the database transaction log. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only. By default, the database transaction log is stored in the same directory as the directory entries themselves, `<NSHOME>/slapd-<serverID>/db`. For fault-tolerance and performance reasons you may want to move this log file to another physical disk. The db_logdirectory parameter is absent from `slapd.conf`. To change the location of the database transaction log, you add the parameter to `slapd.conf`.

For more information on database transaction logging, see “Managing Database Transaction Logging” on page 93.

Default Value

<NSHOME>/slapd-<serverID>/db

Valid Range

Any valid path and directory name.

Slapd.ldbm.conf Syntax

```
db_logdirectory “<directory name>”
```

Example

```
db_logdirectory “/logs/txnlog”
```

db_home_directory

Solaris only. Used to fix a situation in Solaris where the operating system endlessly flushes pages. This flushing can be so excessive that performance of the entire system is severely degraded.

This situation will occur only for certain combinations of the database cache size, the size of physical memory, and kernel tuning parameters. In particular, this situation should not occur if the database cache size is less than 100mb.

If your Solaris host seems excessively slow and your database cache size is around 100mb or more, then you can use the iostat utility to diagnose the problem. Use iostat to monitor the activity of the disk where the directory server's database files are stored. If all of the following conditions are true, then you should set the db_home_directory parameter:

- The disk is heavily used (more than 1mb per second of data transfer)
- There is a long service time (more than 100ms)
- There is mostly write activity

Note The directory referenced by db_home_directory must be a subdirectory of a filesystem of type tempfs (such as /tmp). However, the directory server does not create the subdirectory referenced by this parameter. You must create the directory either manually or by using a script. Failure to create the directory referenced on the db_home_directory parameter will result in the directory server being unable to start.

Also, if you have multiple directory server's on the same machine, their db_home_directory parameters must be configured with different directories. Failure to do so will result in the databases for both directories becoming corrupted.

Finally, use of this parameter causes internal directory server database files to be moved to the directory referenced by the parameter. It is possible, but unlikely, that the server will no longer start after the files have been moved because enough memory cannot be committed. This is a symptom of an overly large database cache size being configured for your server. If this happens, reduce the size of your database cache size to a value where the server will start again.

Default value

N/A

Valid range

Any valid directory name in a tempfs filesystem, such as `/tmp`.

slapd.ldbm.conf Syntax

```
db_home_directory /tmp/<subdirectory>
```

Example

```
db_home_directory /tmp/slapd-phonebook
```

Look Through Limit

Specifies the maximum number of entries that the directory server will check when seeking candidate entries in response for a search request. If this limit is reached, the server returns any entries it has located that match the search request, as well as an exceeded size limit error. For a general discussion of the searching algorithm, refer to “The Searching Algorithm” on page 176.

A null string on this parameter causes no limit to be used; the directory server will check every candidate entry it can find. To set this no limit value from within `slapd.conf`, specify a negative value on the parameter. A value of zero (0) causes no candidate entries to be checked for searches.

Default value

5000

Valid range

-1 to 65535

A value of -1 on this parameter in `slapd.conf` is the same as leaving the parameter blank in the server console; it causes no limit to be used. However, you cannot specify a negative integer for this field in the server console, nor can you specify a null value in `slapd.conf`.

slapd.ldbm.conf Syntax

```
lookthroughlimit <integer>
```

Example

```
lookthroughlimit 5000
```

Maximum Cache Size

Specifies the size in bytes of the in-memory cache. Increasing this number uses more memory but can substantially improve server performance, especially during modifications or when the indexes are being built. Do not increase this number beyond the available resources for your machine.

For more information on this parameter, see the Entry cache hit Ratio field description in “Summary Information Table” on page 281.

Default value

10000000

Valid range

1 to maximum integer

slapd.ldbm.conf Syntax

```
dbcachesize <integer>
```

Example

```
dbcachesize 10000000
```

Maximum Entries in Cache

Specifies the number of entries the directory server will maintain in cache. Increasing this number uses more memory but can substantially improve search performance. The actual amount of memory required per additional entry

depends on the nature of the data stored within the directory server. However, as a general guideline, you can estimate that each entry maintained in cache requires approximately 1 KB (1024 bytes) of memory.

For more information on this parameter, see the Entry cache hit Ratio field description in “Summary Information Table” on page 281.

Default value

1000

Valid range

1 to the total number of database entries.

slapd.ldbm.conf Syntax

```
cachesize <integer>
```

Example

```
cachesize 1000
```

Mode

Specifies the permissions used for newly created index files. This parameter is not available from the server console.

Default value

0600

Valid range

Any four-digit octal number. However, mode 0600 is recommended. This allows read and write access for the owner of the index files (which is the user that `ns-slapd` runs as), and no access for other users.

slapd.ldbm.conf Syntax

```
mode <protection mode>
```

Example

```
mode 0600
```

Read-only

Specifies whether the database is in read-only mode. Any attempt to modify a database in read-only mode returns an error indicating that the server is unwilling to perform the operation.

Default value

off

Valid range

on | off

slapd.ldbm.conf Syntax

```
readonly <Boolean>
```

Example

```
readonly off
```

Suffix

Specifies the distinguished name suffix used for the local database. Incoming queries must have a suffix matching this value. Queries for entries using a suffix other than the value specified in this parameter will be referred to the LDAP server identified on the Referral parameter.

Multiple suffixes can be configured for your local database if multiple root points are used in your database. Two suffixes always exist for a directory server database. The first is the suffix you configure when you initially install the directory server, and this suffix represents your directory tree's root point. The second suffix is used for machine data. See "Machine data" on page 356 for more information.

A suffix must always be set for your directory tree in order for clients to successfully access the tree.

For information on setting suffixes for your directory, see “Setting Suffixes for Your Database” on page 85.

Valid range

Any valid distinguished name.

slapd.ldbm.conf Syntax

suffix <string>

Example

```
suffix “o=airius.com”
```

If the suffix DN contains a comma, the comma must be escaped by a single backslash (on NT) or double backslashes (on Unix). For example, to set a suffix of Airius Bolivia, S.A., you would enter

```
suffix “o=Airius Bolivia\, S.A.”
```

on NT or

```
suffix “o=Airius Bolivia\\, S.A.”
```

on Unix.

Appendix

A

LDAP URLs

An LDAP URL is a URL that begins with the `ldap://` protocol prefix (or `ldaps://`, if the server is communicating over an SSL connection) and specifies a search request sent to an LDAP server.

When you access the directory server using a web-based client such as the directory server gateway, you must provide an LDAP URL identifying the directory server you wish to access. You can set the default LDAP URL to use with the directory server gateway using the `baseurl` parameter (in the gateway configuration file).

In addition, you may use LDAP URLs when managing directory server referrals or access control instructions.

This appendix contains the following sections:

Components of an LDAP URL

LDAP URLs have the following syntax:

```
ldap[s]://<hostname>:<port>/<base_dn>?<attributes>?<scope>?<filter>
```

The `ldap://` protocol is used to connect to LDAP servers over unsecured connections, and the `ldaps://` protocol is used to connect to LDAP servers over SSL connections. Table A.1 lists the components of an LDAP URL.

Table A.1 Components of an LDAP URL

Component	Description
`<hostname>`	Name (or IP address in dotted format) of the LDAP server (for example, `ldap.airius.com` or `192.202.185.90`).
`<port>`	Port number of the LDAP server (for example, 696). If no port is specified, the standard LDAP port (389) or LDAPS port (636) is used.
`<base_dn>`	Distinguished name (DN) of an entry in the directory. This DN identifies the entry that is starting point of the search. If this component is empty, the search starts at the root of the directory tree.
`<attributes>`	The attributes to be returned. To specify more than one attribute, use commas to delimit the attributes (for example, `"cn,mail,telephoneNumber"`). If no attributes are specified in the URL, all attributes are returned.

Table A.1 Components of an LDAP URL (Continued)

Component	Description
<scope>	The scope of the search, which can be one of these values: • base retrieves information only about the distinguished name (<base_dn>) specified in the URL. • one retrieves information about entries one level below the distinguished name (<base_dn>) specified in the URL. The base entry is not included in this scope. • sub retrieves information about entries at all levels below the distinguished name (<base_dn>) specified in the URL. The base entry is included in this scope. If no scope is specified, the server performs a base search.
<filter>	Search filter to apply to entries within the specified scope of the search. If no filter is specified, the server uses the filter (objectClass=*).

<attributes>, <scope>, and <filter> are identified by their positions in the URL. If you do not want to specify any attributes, you still need to include the question marks delimiting that field.

For example, to specify a subtree search starting from "o=airius.com" that returns all attributes for entries matching "(sn=Jensen)", use the following URL:

```
ldap://ldap.airius.com/o=airius.com??sub?(sn=Jensen)
```

The two consecutive question marks—??—indicate that no attributes have been specified. Since no specific attributes are identified in the URL, all attributes are returned in the search.

Escaping Unsafe Characters

Any “unsafe” characters in the URL need to be represented by a special sequence of characters (this is often called escaping unsafe characters). For example, a space is an unsafe character that must be represented as `%20` within the URL. Thus, the distinguished name `"o=airius corporation"` must be encoded as `"o=airius%20corporation"`. The following table lists the characters that are considered unsafe within URLs and provides the associated escape characters to use in place of the unsafe character:

Unsafe character	Escape characters
space	%20
<	%3c
>	%3e
"	%22
#	%23
%	%25
{	%7b
}	%7d
|	%7c
\	%5c
^	%5e
~	%7e
[	%5b
]	%5d
‘	%60

Examples of LDAP URLs

The following LDAP URL specifies a base search for the entry with the distinguished name "o=airius.com".

```
ldap://ldap.airius.com/o=airius.com
```

- Because no port number is specified, the standard LDAP port number (389) is used.
- Because no attributes are specified, the search returns all attributes.
- Because no search scope is specified, the search is restricted to the base entry "o=airius.com".
- Because no filter is specified, the default filter "(objectclass=*)" is used.

The following LDAP URL retrieves the postalAddress attribute of the airius.com entry:

```
ldap://ldap.airius.com/o=airius.com?postalAddress
```

- Because no search scope is specified, the search is restricted to the base entry "o=airius.com".
- Because no filter is specified, the default filter "(objectclass=*)" is used.

The following LDAP URL retrieves the cn, mail, and telephoneNumber attributes of the entry for Barbara Jensen:

```
ldap://ldap.airius.com/cn=Barbara%20Jensen,o=airius.com?cn,mail,telepho
neNumber
```

- Because no search scope is specified, the search is restricted to the base entry "cn=Barbara Jensen,o=airius.com".
- Because no filter is specified, the default filter "(objectclass=*)" is used.

The following LDAP URL specifies a search for entries that have the last name `Jensen` and are at any level under `"o=airius.com"`:

```
ldap://ldap.airius.com/o=airius.com??sub?(sn=Jensen)
```

- Because no attributes are specified, the search returns all attributes.
- Because the search scope is `sub`, the search encompasses the base entry `"o=airius.com"` and entries at all levels under the base entry.

The following LDAP URL specifies a search for the object class for all entries one level under `"o=airius.com"`:

```
ldap://ldap.airius.com/o=airius.com?objectClass?one
```

- Because the search scope is `one`, the search encompasses all entries one level under the base entry `"o=airius.com"`. The search scope does not include the base entry.
- Because no filter is specified, the default filter `"(objectclass=*)"` is used.

Important The syntax for LDAP URLs does not include any means for specifying credentials or passwords. Search requests initiated through LDAP URLs are unauthenticated, unless the LDAP client that supports LDAP URLs provides for authentication. The Netscape Directory Server gateway supports this.

Appendix

B

Internationalization

The directory server allows you to store, manage, and search for entries and their associated attributes in a number of different languages. An internationalized directory can be an invaluable corporate resource in that it provides employees and business partners with immediate access to the information they need in the languages they can understand.

The directory supports all international characters sets by default because directory data is stored in UTF-8. Further, the directory server allows you to specify search matching rules and collation orders based on language preferences.

Note You must use ASCII characters for attribute and object class names. For information on object classes, attributes, and the directory server schema, see the *Netscape Directory Server Deployment Manual*. For information on extending your schema, see Chapter 3, "Extending the Directory Schema."

The directory server provides support for multiple languages through the use of locales. A locale identifies language-specific information about how users of a specific region, culture, and/or custom expect data to be presented, including how data of a given language is interpreted and how data is to be sorted, or collated. In addition, the locale information indicates what code page should be used to represent a given language. A code page is simply an internal table that the operating system uses to relate keyboard keys to character font screen displays.

More specifically, a locale specifies:

- Collation order

  The collation order provides language and cultural-specific information about how the characters of a given language are to be sorted. It identifies things like the sequence of the letters in the alphabet, how to compare letters with accents with letters without accents, and if there are any characters that can be ignored when comparing strings. The collation order also takes into account culture-specific information about a language such as the direction in which it is read (left to right, right to left, or up and down). The directory server uses the collation orders to perform searches on internationalized directory data.

- Character type

  The character type distinguishes alphabetic characters from numeric or other characters. In addition, it defines the mapping of upper-case to lower-case letters. For example, in some languages, the | character is considered punctuation while in others it is considered alphabetic.

- Monetary format

  The monetary format specifies the monetary symbol used by a specific region, whether the symbol goes before or after its value, and how monetary units are represented.

- Time/date format

  The time and date format indicates the customary formatting for times and dates in the region. The time/date format indicates such things as whether dates are customarily represented in the mm/dd/yy, month day, year, or dd/mm/yy format and specifies what the days of the week and month are in a given language. For example, the date January 10, 1996 is represented as 10.leden 1996 in Czechoslovakian and 10 janvier 1996 in French.

Because a locale takes into account cultural, customary, and regional differences in addition to mechanical language differences, the directory data can both be translated into the specific languages understood by your users as well as be presented in a way that users in a given region expect.

Locale information is automatically copied to the `<NSHOME>/lib/nls/locale30` directory during the directory server installation.

Identifying Supported Locales

When performing directory server operations that require you to specify a locale, such as a search operation, you can either use a language tag or a collation order object identifier (OID).

A language tag is a string that begins with the two-character lowercase language code that identifies the language (as defined in ISO standard 639). If necessary to distinguish regional differences in language, the language tag may also contain a country code, which is a two-character string (as defined in ISO standard 3166). The language code and country code are separated by a hyphen. For example, the language tag used to identify the British English locale is en-GB.

An object identifier (OID) is a decimal number used to uniquely identify an object, such as an attribute or object class, in an object-oriented system such as the directory server. The OIDs you use when searching or indexing an internationalized directory identify specific collation orders supported by the directory server. For example, the OID `2.16.840.1.113730.3.3.2.17.1` identifies the Finnish collation order.

When performing an international search on the directory server data, you can use either the language tag or the OID to identify the collation order you want to use. However, when setting up an international index, you must use the OIDs.

The following table lists each locale supported by the directory server and identifies the associated language tags and OIDs.

Table B.1 Supported locales

Locale	Language tag	Collation order object identifiers (OIDs)
Albanian	`sq`	`2.16.840.1.113730.3.3.2.44.1`
Arabic	`ar`	`2.16.840.1.113730.3.3.2.1.1`
Byelorussian	`be`	`2.16.840.1.113730.3.3.2.2.1`
Bulgarian	`bg`	`2.16.840.1.113730.3.3.2.3.1`
Catalan	`ca`	`2.16.840.1.113730.3.3.2.4.1`
Chinese (Simplified)	`zh`	`2.16.840.1.113730.3.3.2.49.1`
Chinese (Traditional)	`zh-TW`	`2.16.840.1.113730.3.3.2.50.1`

Table B.1 Supported locales (Continued)

Locale	Language tag	Collation order object identifiers (OIDs)
Croatian	hr	2.16.840.1.113730.3.3.2.22.1
Czechoslovakian	cs	2.16.840.1.113730.3.3.2.5.1
Danish	da	2.16.840.1.113730.3.3.2.6.1
English (US)	en or en-US	2.16.840.1.113730.3.3.2.11.1
Estonian	et	2.16.840.1.113730.3.3.2.16.1
Finnish	fi	2.16.840.1.113730.3.3.2.17.1
French	fr or fr-FR	2.16.840.1.113730.3.3.2.18.1
German	de	2.16.840.1.113730.3.3.2.7.1
Greek	el	2.16.840.1.113730.3.3.2.10.1
Hebrew	iw	2.16.840.1.113730.3.3.2.27.1
Hungarian	hu	2.16.840.1.113730.3.3.2.23.1
Icelandic	is	2.16.840.1.113730.3.3.2.24.1
Japanese	ja	2.16.840.1.113730.3.3.2.28.1
Korean	ko	2.16.840.1.113730.3.3.2.29.1
Latvian, Lettish	lv	2.16.840.1.113730.3.3.2.31.1
Lithuanian	lt	2.16.840.1.113730.3.3.2.30.1
Macedonian	mk	2.16.840.1.113730.3.3.2.32.1
Norwegian	no	2.16.840.1.113730.3.3.2.35.1
Polish	pl	2.16.840.1.113730.3.3.2.38.1
Romanian	ro	2.16.840.1.113730.3.3.2.39.1
Russian	ru	2.16.840.1.113730.3.3.2.40.1
Serbian (Cyrilic)	sr	2.16.840.1.113730.3.3.2.45.1
Serbian (Latin)	sh	2.16.840.1.113730.3.3.2.41.1
Slovakian	sk	2.16.840.1.113730.3.3.2.42.1

Table B.1 Supported locales (Continued)

Locale	Language tag	Collation order object identifiers (OIDs)
Slovenian	sl	2.16.840.1.113730.3.3.2.43.1
Spanish	es or es-ES	2.16.840.1.113730.3.3.2.15.1
Swedish	sv	2.16.840.1.113730.3.3.2.46.1
Turkish	tr	2.16.840.1.113730.3.3.2.47.1
Ukranian	uk	2.16.840.1.113730.3.3.2.48.1

For information on how to set up a Netscape client to use non-English character sets, refer to the *Netscape Directory Server Installation Guide*.

Supported Language Subtypes

The following table contains the list of supported language subtypes. Language subtypes can be used by clients to determine which specific values for which to search. For more information on using language subtypes, see “Adding an Attribute Subtype Using the Property Editor” on page 237.

Table B.2 Supported Language Subtypes

Language tag	Language
af	Afrikaans
be	Byelorussian
bg	Bulgarian
ca	Catalan
cs	Czechoslovakian
da	Danish
de	German
el	Greek
en	English

Table B.2 Supported Language Subtypes (Continued)

Language tag	Language
es	Spanish
eu	Basque
fi	Finnish
fo	Faroese
fr	French
ga	Irish
gl	Galician
hr	Croatian
hu	Hungarian
id	Indonesian
is	Icelandic
it	Italian
ja	Japanese
ko	Korean
nl	Dutch
no	Norwegian
pl	Polish
pt	Portuguese
ro	Romanian
ru	Russian
sk	Slovakian
sl	Slovenian
sq	Albanian
sr	Serbian
sv	Swedish
tr	Turkish

Table B.2 Supported Language Subtypes (Continued)

Language tag	Language
uk	Ukrainian
zh	Chinese

Glossary

access control list *See ACL.*

ACL Access control list. Netscape's mechanism for controlling access to your directory.

attribute Holds descriptive information about an entry. Attributes have a label and a value. Each attribute also follows a standard syntax for the type of information that can be stored as the attribute value.

attribute list A list of required and optional attributes for a given entry type or object class.

authentication 1. Process of proving the identity of the client user to the Directory Server. Users must provide a bind DN and the corresponding password in order to be granted access to the directory. The Directory Server allows the user to perform functions or access files and directories based on the permissions granted to that user by the directory administrator.

2. Allows a client to make sure they are connected to a secure server, preventing another computer from impersonating the server or attempting to appear secure when it is not.

authentication certificate Digital file that is not transferable and not forgeable and is issued by a third party. Authentication certificates are sent from server to client or client to server in order to verify and authenticate the other party.

bind DN Distinguished name used to authenticate to the Directory Server when performing an operation.

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 Certification Authority.*

Certification Authority Company or organization that sells and issues authentication certificates. You may purchase an authentication certificate from a Certification Authority that you trust. Also known as a CA.

CGI Common Gateway Interface. An interface for external programs to communicate with the HTTP server. Programs written to use CGI are called CGI programs or CGI scripts, and can be written in many of the common programming languages. CGI programs handle forms or perform output parsing that is not done by the server itself.

ciphertext Encrypted information that cannot be read by anyone without the proper key to decrypt the information.

CIR *See consumer-initiated replication*

client *See LDAP client.*

consumer Server containing replicated directory trees or subtrees from a supplier server.

consumer-initiated replication Replication configuration where consumer servers pull directory data from supplier servers.

daemon A background process on a Unix machine that is responsible for a particular system task. Daemon processes do not need human intervention to continue functioning.

Directory Server gateway A collection of CGI forms that allows a browser to perform LDAP client functions, such as querying and accessing a Directory Server, from a web browser.

directory service A database application designed to manage descriptive, attribute-based information about people and resources within an organization.

distinguished name String representation of an entry's name and location in an LDAP directory.

DNS Domain Name System. The system used by machines on a network to associate standard IP addresses (such as 198.93.93.10) with hostnames (such as www.netscape.com). Machines normally get the IP address for a hostname from a DNS server, or they look it up in tables maintained on their systems.

DNS alias A DNS alias is a hostname that the DNS server knows points to a different host—specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, an alias such as `www.[yourdomain].[domain]` might point to a real machine called `realthing.[yourdomain].[domain]` where the server currently exists.

file extension The section of a filename after the period or dot (.) that typically defines the type of file (for example, .GIF and .HTML). In the filename `index.html` the file extension is `html`.

file type The format of a given file. For example, graphics files are often saved in GIF format, while a text file is usually saved as ASCII text format. File types are usually identified by the file extension (for example, .GIF or .HTML).

gateway *See Directory Server gateway*.

hostname A name for a machine in the form machine.domain.dom, which is translated into an IP address. For example, www.netscape.com is the machine www in the subdomain netscape 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 the 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.

HTTPD An abbreviation for the HTTP daemon or service, a program that serves information using the HTTP protocol. The daemon or service is often called an httpd.

HTTP-NG The next generation of Hypertext Transfer Protocol.

HTTPS A secure version of HTTP, implemented using the Secure Sockets Layer, SSL.

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, 198.93.93.10).

LDAP Lightweight Directory Access Protocol. Directory service protocol designed to run over TCP/IP and across multiple platforms.

LDAP client Software used to request and view LDAP entries from an LDAP Directory Server. See also browser.

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 *See LDAP.*

management information base *See MIB.*

master agent *See SNMP master agent.*

MD5 A message digest algorithm by RSA Data Security, Inc., which can be used to produce a short digest of data, that is unique with high probability, and is mathematically extremely hard to produce a piece of data that will produce the same message digest.

MD5 signature A message digest produced by the MD5 algorithm.

MIB Management Information Base.

network management application An application on a network that can be managed by SNMP.

network management station *see NMS.*

NIS Network Information Service. A system of programs and data files that Unix machines use to collect, collate, and share specific information about machines, users, file systems, and network parameters throughout a network of computers.

NMS Network Management Station.

ns-slapd Netscape's LDAP Directory Server daemon or service that is responsible for all actions of the Directory Server. See also slapd.

object class Defines an entry type in the directory by defining which attributes are contained in the entry.

object identifier A string, usually of decimal numbers, that uniquely identifies an object, such as an object class or an attribute, in an object-oriented system.

OID *See object identifier.*

password file A file on Unix machines that stores Unix user login names, passwords, and user ID numbers. It is also known as `/etc/passwd`, because of where it is kept.

PDU Protocol Data Unit.

protocol A set of rules that describes how devices on a network exchange information.

protocol data units *See PDU.*

proxy DN Used with proxied authorization. The proxy DN is the DN of an entry that has access permissions to the target on which the client-application is attempting to perform an operation.

public-key encryption Encryption that uses two keys: a public key for encrypting data, and a private key for decrypting data. Someone sending you encrypted information encrypts it using your public key. The information can then only be decrypted using your private key.

RAM Random access memory. The physical semiconductor-based memory in a computer. Information stored in RAM is lost when the computer is shut down.

rc.local A file on Unix machines that describes programs that are run when the machine starts. It is also called `/etc/rc.local` because of its location.

RDN Relative distinguished name. The name of the actual entry itself, before the entry's ancestors have been appended to the string to form the full distinguished name.

referential integrity Mechanism that ensures that relationships between related entries are maintained within the directory.

replication Act of copying directory trees or subtrees from supplier servers to consumer servers.

replication agreement Set of configuration parameters that identify the directory objects to replicate, the times during which replication can occur, and the servers involved in the replication process.

RFC Request For Comments. Procedures or standards documents submitted to the Internet community. People can send comments on the technologies before they become accepted standards.

root The most privileged user available on Unix machines. The root user has complete access privileges to all files on the machine.

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 entries added or modified in the directory conform to the defined schema. Schema checking is on by default and users will receive an error if they try to save an entry that does not conform to the schema.

Secure Sockets Layer *See SSL*.

Server Console Java-based application that allows you to perform administrative management of your Directory Server from a GUI.

server daemon The server daemon is a process that, once running, listens for and accepts requests from clients.

server service The server service is a process on Windows NT that, once running, listens for and accepts requests from clients.

server root A directory on the server machine dedicated to holding the server program and configuration, maintenance, and information files.

Server Selector Interface that allows you select and configure servers using a browser.

service A background process on a Windows NT machine that is responsible for a particular system task. Service processes do not need human intervention to continue functioning.

SIR *See supplier-initiated replication*.

slapd LDAP Directory Server daemon or service that is responsible for most functions of a directory except replication. See also ns-slapd.

SNMP master agent Software that exchanges information between the various subagents and the NMS.

SNMP subagent Software that gathers information about the managed device and passes the information to the master agent.

SSL Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

subagent *See SNMP subagent.*

superuser The most privileged user available on Unix machines (also called root). The superuser has complete access privileges to all files on the machine.

supplier Server containing the master copy of directory trees or subtrees that are replicated to consumer servers.

supplier-initiated replication Replication configuration where supplier servers replicate directory data to consumer servers.

symmetric encryption Encryption that uses the same key for both encrypting and decrypting.

TCP/IP Transmission Control Protocol/Internet Protocol. The main network protocol for the Internet and for enterprise (company) networks.

uid A unique number associated with each user on a Unix system.

URL Uniform Resource Locator. The addressing system used by the server and the client to request documents. It 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, freeing the user of having to place it in the URL.

A sample URL is `http://www.netscape.com/index.html`.

X.500 standard The set of ISO/IEC documents outlining the standard object classes, attributes, and LDAP protocols to be used in directory server creation and management.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X