Sun Directory Server Enterprise Edition 7.0 Administration Guide

Part No: 820–4809
November 2009

100226@23474

Contents

Preface

The *Administration Guide*, provides procedural information for configuring Directory Server and Directory Proxy Server features from the command line. Instructions for configuring these feature by using the web-based interface (Directory Service Control Center) are provided in the online help.

Who Should Use This Book

This *Administration Guide* is intended for administrators of Directory Server and Directory Proxy Server software.

Before You Read This Book

This book does not provide information on installing the software. For installation information, see the *Sun Directory Server Enterprise Edition 7.0 Installation Guide*.

If you are migrating from an older version of Directory Server or Directory Proxy Server, see the *Sun Directory Server Enterprise Edition 7.0 Upgrade and Migration Guide* for instructions on migrating servers. If you are unfamiliar with the new features in this version, it might be useful to read the *Sun Directory Server Enterprise Edition 7.0 Evaluation Guide* for an overview of the new features.

How This Book Is Organized

Part I, "Directory Server Administration," provides procedural information on administering Directory Server.

Part II, "Directory Proxy Server Administration," provides procedural information on administering Directory Proxy Server

Examples Used in This Guide

For consistency reasons, the same example data is used throughout this guide. Replace these values with the appropriate values for your system.

TABLE P–1 Default Values Used in Examples

Variable	Values used in examples
Suffix (SUFFIX_DN)	dc=example,dc=com
Instance path (INSTANCE_PATH)	For Directory Server: /local/dsInst/ For Directory Proxy Server: /local/dps/
Hostnames (HOST)	host1, host2, host3
Port (PORT)	LDAP: Default for root: 389. Default for non-root: 1389 SSL default: Default for root: 636. Default for non-root: 1636

Sun Directory Server Enterprise Edition Documentation Set

This documentation set explains how to use Sun™ Directory Server Enterprise Edition to evaluate, design, deploy, and administer directory services. In addition, it shows how to develop client applications for Directory Server Enterprise Edition. The Directory Server Enterprise Edition documentation set is available at `http://docs.sun.com/coll/1819.1`.

The following table lists all the available documents.

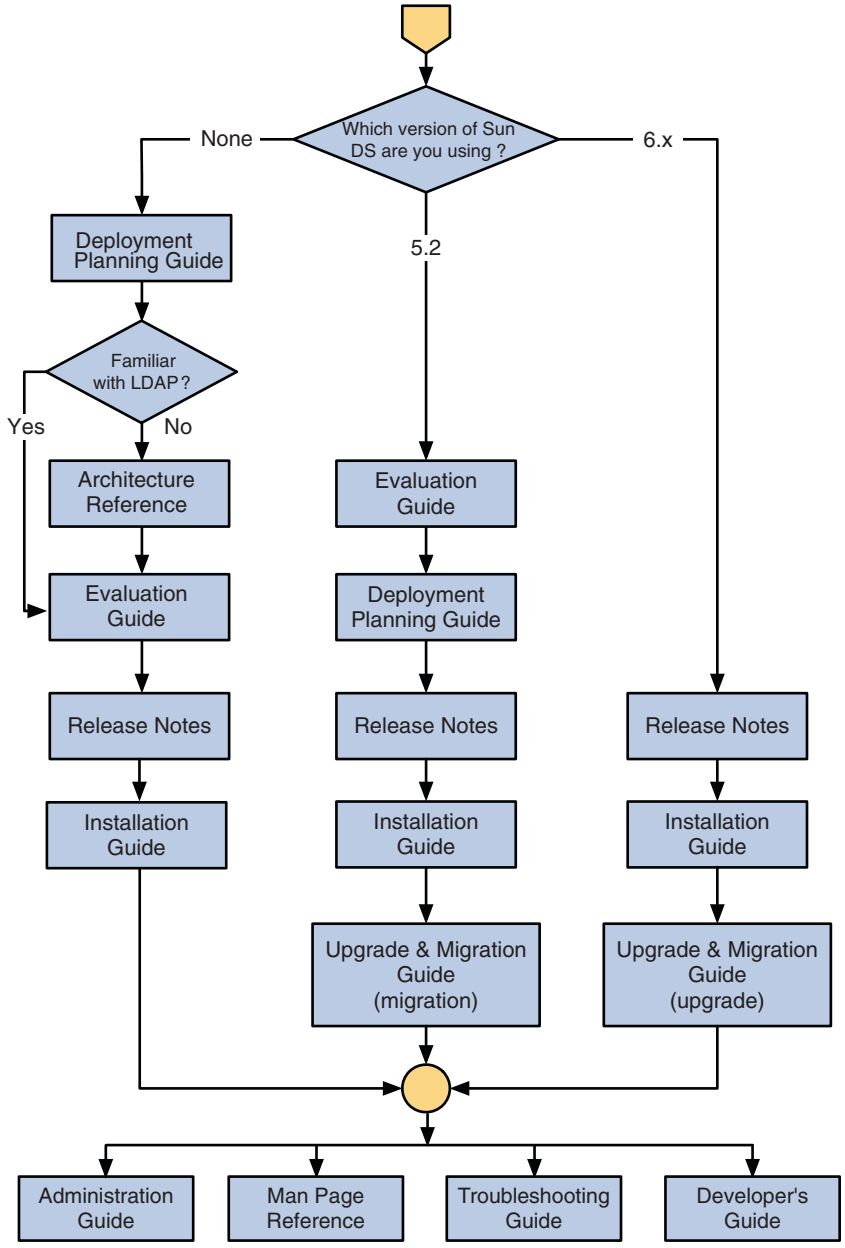
TABLE P–2 Directory Server Enterprise Edition Documentation

Document Title	Contents
Sun Directory Server Enterprise Edition 7.0 Release Notes	Contains the latest information about Directory Server Enterprise Edition, including known problems.
Sun Directory Server Enterprise Edition 7.0 Documentation Center	Contains links to key areas of the documentation set that help you to quickly locate the key information.
Sun Directory Server Enterprise Edition 7.0 Evaluation Guide	Introduces the key features of this release. Demonstrates how these features work and what they offer in the context of a deployment that you can implement on a single system.

TABLE P–2 Directory Server Enterprise Edition Documentation *(Continued)*

Document Title	Contents
Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide	Explains how to plan and design highly available, highly scalable directory services based on Directory Server Enterprise Edition. Presents the basic concepts and principles of deployment planning and design. Discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Directory Server Enterprise Edition.
Sun Directory Server Enterprise Edition 7.0 Installation Guide	Explains how to install the Directory Server Enterprise Edition software. Shows how to configure the installed software and verify the configured software.
Sun Directory Server Enterprise Edition 7.0 Upgrade and Migration Guide	Provides upgrade instructions to upgrade the version 6 installation and migration instructions to migrate version 5.2 installations.
Sun Directory Server Enterprise Edition 7.0 Administration Guide	Provides command-line instructions for administering Directory Server Enterprise Edition. For hints and instructions about using the Directory Service Control Center, DSCC, to administer Directory Server Enterprise Edition, see the online help provided in DSCC.
Sun Directory Server Enterprise Edition 7.0 Developer's Guide	Shows how to develop directory client applications with the tools and APIs that are provided as part of Directory Server Enterprise Edition.
Sun Directory Server Enterprise Edition 7.0 Reference	Introduces technical and conceptual foundations of Directory Server Enterprise Edition. Describes its components, architecture, processes, and features.
Sun Directory Server Enterprise Edition 7.0 Man Page Reference	Describes the command-line tools, schema objects, and other public interfaces that are available through Directory Server Enterprise Edition. Individual sections of this document can be installed as online manual pages.
Sun Directory Server Enterprise Edition 7.0 Troubleshooting Guide	Provides information for defining the scope of the problem, gathering data, and troubleshooting the problem areas by using various tools.
Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide	Provides general guidelines and best practices for planning and deploying Identity Synchronization for Windows.
Sun Java System Identity Synchronization for Windows 6.0 Installation and Configuration Guide	Describes how to install and configure Identity Synchronization for Windows.
Additional Installation Instructions for Sun Java System Identity Synchronization for Windows 6.0	Provides additional installation instructions in context of Directory Server Enterprise Edition 7.0.

For an introduction to Directory Server Enterprise Edition, review the following documents in the order in which they are listed.



Related Reading

The SLAMD Distributed Load Generation Engine is a Java™ application that is designed to stress test and analyze the performance of network-based applications. This application was

originally developed by Sun Microsystems, Inc. to benchmark and analyze the performance of LDAP directory servers. SLAMD is available as an open source application under the Sun Public License, an OSI-approved open source license. To obtain information about SLAMD, go to `http://www.slamd.com/`. SLAMD is also available as a java.net project. See `https://slamd.dev.java.net/`.

Java Naming and Directory Interface (JNDI) supports accessing the Directory Server using LDAP and DSML v2 from Java applications. For information about JNDI, see `http://java.sun.com/products/jndi/`. The *JNDI Tutorial* contains detailed descriptions and examples of how to use JNDI. This tutorial is at `http://java.sun.com/products/jndi/tutorial/`.

Directory Server Enterprise Edition can be licensed as a standalone product, as part of a suite of Sun products, such as the Sun Java Identity Management Suite, or as an add-on package to other software products from Sun.

Identity Synchronization for Windows uses Message Queue with a restricted license. Message Queue documentation is available at `http://docs.sun.com/coll/1307.2`.

Identity Synchronization for Windows works with Microsoft Windows password policies.

- Information about password policies for Windows 2003, is available in the Microsoft documentation online.
- Information about the Microsoft Certificate Services Enterprise Root certificate authority, is available in the Microsoft support documentation online.
- Information about configuring LDAP over SSL on Microsoft systems, is available in the Microsoft support documentation online.

Redistributable Files

Directory Server Enterprise Edition does not provide any files that you can redistribute.

Default Paths and Command Locations

This section explains the default paths used in documentation, and provides locations of commands on different operating systems and deployment types.

Default Paths

The table in this section describes the default paths that are used in this document. For complete descriptions of the files installed, see Chapter 1, "Directory Server Enterprise Edition File Reference," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

TABLE P–3 Default Paths

Placeholder	Description	Default Value
install-path	Represents the base installation directory for Directory Server Enterprise Edition software.	When you install from a zip distribution using unzip, the *install-path* is the *current-directory*`/dsee7`. When you install from a native package distribution, the default *install-path* is `/opt/SUNWdsee7`.
instance-path	Represents the full path to an instance of Directory Server or Directory Proxy Server. Documentation uses `/local/dsInst/` for Directory Server and `/local/dps/` for Directory Proxy Server.	No default path exists. Instance paths must nevertheless always be found on a *local* file system. On Solaris systems, the `/var` directory is recommended:
serverroot	Represents the parent directory of the Identity Synchronization for Windows installation location	Depends on your installation. Note that the concept of a *serverroot* no longer exists for Directory Server and Directory Proxy Server.
`isw-`*hostname*	Represents the Identity Synchronization for Windows instance directory	Depends on your installation
/path/to/`cert8.db`	Represents the default path and file name of the client's certificate database for Identity Synchronization for Windows	*current-working-dir*`/cert8.db`
serverroot`/isw-`*hostname*`/logs/`	Represents the default path to the Identity Synchronization for Windows local log files for the System Manager, each connector, and the Central Logger	Depends on your installation
serverroot`/isw-`*hostname*`/logs/central/`	Represents the default path to the Identity Synchronization for Windows central log files	Depends on your installation

Command Locations

The table in this section provides locations for commands that are used in Directory Server Enterprise Edition documentation. To learn more about each of the commands, see the relevant man pages.

TABLE P–4 Command Locations

Command	Native Package Distribution	Zip Distribution
`cacaoadm`	`/usr/sbin/cacaoadm`	Solaris, Linux, HP—UX — *install-path*`/bin/cacaoadm`
		Windows - *install-path*`\bin\cacaoadm.bat`
`certutil`	`/usr/sfw/bin/certutil`	*install-path*`/bin/certutil`
`dpadm`(1M)	*install-path*`/bin/dpadm`	*install-path*`/bin/dpadm`
`dpconf`(1M)	*install-path*`/bin/dpconf`	*install-path*`/bin/dpconf`
`dsadm`(1M)	*install-path*`/bin/dsadm`	*install-path*`/bin/dsadm`
`dsccmon`(1M)	*install-path*`/bin/dsccmon`	*install-path*`/bin/dsccmon`
`dsccreg`(1M)	*install-path*`/bin/dsccreg`	*install-path*`/bin/dsccreg`
`dsccsetup`(1M)	*install-path*`/bin/dsccsetup`	*install-path*`/bin/dsccsetup`
`dsconf`(1M)	*install-path*`/bin/dsconf`	*install-path*`/bin/dsconf`
`dsmig`(1M)	*install-path*`/bin/dsmig`	*install-path*`/bin/dsmig`
`dsutil`(1M)	*install-path*`/bin/dsutil`	*install-path*`/bin/dsutil`
`entrycmp`(1)	*install-path*`/bin/entrycmp`	*install-path*`/bin/entrycmp`
`fildif`(1)	*install-path*`/bin/fildif`	*install-path*`/bin/fildif`
`idsktune`(1M)	Not provided	At the root of the unzipped zip distribution
`insync`(1)	*install-path*`/bin/insync`	*install-path*`/bin/insync`
`ldapsearch`(1)	`/opt/SUNWdsee/dsee6/bin`	*install-path*`/dsrk/bin`
`repldisc`(1)	*install-path*`/bin/repldisc`	*install-path*`/bin/repldisc`

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–5 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–6 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`

Shell Prompts in Command Examples

The following table shows default system prompts and superuser prompts.

TABLE P–7 Shell Prompts

Shell	Prompt
C shell on UNIX and Linux systems	`machine_name%`
C shell superuser on UNIX and Linux systems	`machine_name#`
Bourne shell and Korn shell on UNIX and Linux systems	`$`
Bourne shell and Korn shell superuser on UNIX and Linux systems	`#`
Microsoft Windows command line	`C:\`

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–8 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Feedback.

PART I

Directory Server Administration

◆ ◆ ◆ CHAPTER 1

Directory Server Tools

Sun Directory Server provides a browser interface and command-line tools for administering multiple servers, instances, and suffixes in a replicated environment. This chapter provides overview information about Directory Server administration tools.

This chapter covers the following topics:

Directory Server Administration Overview

Information about the Directory Server administration framework is provided in other guides in this documentation set.

- For an overview of the Directory Server administration framework, see "Directory Server Enterprise Edition Administration Model" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.
- For more detailed reference information about the Directory Server administration framework, see Chapter 2, "Directory Server Overview," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

Deciding When to Use DSCC and When to Use the Command Line

Directory Server Enterprise Edition provides two user interfaces for managing Directory Servers and Directory Proxy Servers: a browser interface, Directory Service Control Center (DSCC), and a command-line interface.

Determining Whether a Procedure Can Be Done Using DSCC

Most procedures in this guide can be performed using either the command line or DSCC. The procedures in this guide show how to use the command line to accomplish the procedure. In most cases DSCC can be used to perform the same task. If DSCC can be used for a particular procedure, a statement to that effect appears at the beginning of the procedure.

The DSCC online help provides detailed instructions on how to use DSCC to perform the procedures in this guide.

Cases Where Using DSCC Is Better

DSCC enables you to perform some operations and tasks more easily than you can perform them from the command line, as explained in the following sections. In general, any command that must be applied to several servers is best performed using DSCC.

Viewing Servers and Suffix Replication Status

DSCC displays tables that show all server instances that have been registered in DSCC, all suffixes that have been configured, and the status of each.

The servers table is on the Directory Servers tab and shows the operational status of the server. For a complete list of possible server states, see the Directory Server online help.

The suffixes table is on the Suffixes tab and shows replication status information, such as the number of entries and the number and age of any missing changes. For more information about the information displayed in this table, see the Directory Server online help.

Managing Groups of Servers

Server groups assist you in monitoring and configuring servers. You can create groups and assign servers to the groups. For example, you can group servers by geographical location, or by function. If you have a large number of servers, you can filter the servers shown on the Directory Servers tab so that only the servers in the group are shown. You can also copy the server configuration (for example index or cache settings) of one server to all other servers in a group. For instructions on how to set up and use a server group, see the Directory Server online help.

Copying Configuration Settings

DSCC enables you to copy the configuration settings of an existing server, suffix, or replication agreement to one or more other servers, suffixes, or replication agreements. For information about how to perform each of these tasks, see the Directory Server online help.

Configuring Replication

With DSCC, you can set up a replication topology quickly and easily. Simply create the server instances, then use the steps provided by DSCC to designate the role of each server. DSCC automatically creates the replication agreements for you. For more information about how to configure replication using DSCC, see the Directory Server online help.

Directory Server Command-Line Tools

Most tasks you perform on DSCC can be performed using command-line tools. These tools enable you to manage Directory Server directly from the command line, and to manage your server by using scripts.

The main directory server commands are `dsadm`, `dsconf`, and `dsutil`. You can use these commands to perform backups, export to LDIF, manage certificates, manage the administration of users or roles, and so on. For information about these commands, see the `dsadm(1M)`, `dsconf(1M)`, and `dsutil(1M)` man pages.

The `dsconf`, `dsmig`, `dsccmon`, and `dsutil` are LDAP based commands so you must specify the user bind DN and password for these commands to authenticate. While the `dpadm` and `dsadm` commands operate on the instance files.

This section contains the following information about Directory Server command-line tools:

Location of Directory Server Commands

The Directory Server command-line tools are contained in a default installation directory:

install-path`/bin`

The directory for your installation depends on your operating system. Installation paths for all operating systems are listed in "Default Paths and Command Locations" on page 25.

Setting Environment Variables for `dsconf`

The `dsconf` command requires some options that you can preset by using environment variables. If you do not specify an option when using the command, or do not set the environment variable, the default setting is used. You can configure environment variables for the following options:

`-D` *user DN*	User bind DN. Environment variable: `LDAP_ADMIN_USER`. Default: `cn=Directory Manager`.
`-w` *password-file*	Password file for the user bind DN. Environment variable: `LDAP_ADMIN_PWF`. Default: Prompt for password.
`-h` *host*	Host name. Environment variable: `DIRSERV_HOST`. Default: `local host`.
`-p` *LDAP-port*	LDAP port number. Environment variable: `DIRSERV_PORT`. Default: `389`.
`-e`, `--unsecured`	Specifies that `dsconf` should open a clear connection by default. Environment variable: `DIRSERV_UNSECURED`. If this variable is not set, `dsconf` opens a secure connection by default.

For more details, see the `dsconf(1M)` man page.

Comparison of `dsadm` and `dsconf`

The following table shows a comparison of the `dsadm` and `dsconf` commands.

TABLE 1–1 Comparison of the `dsadm` and `dsconf` Commands

	`dsadm` **Command**	`dsconf` **Command**
Description	Administration commands that must be run directly on the local host. For example: ■ Starting and stopping the server ■ Creating a server instance	Administration commands that can be run from a remote host. For example: ■ Enabling replication ■ Setting cache size
Notes	The server must be stopped (except for the `dsadm stop` and `dsadm info` commands). The server is identified by the server instance path (*instance-path*). You must have OS access permissions to the server instance path.	The server must be running. The server is identified by host name (`-h`) port (`-p`) or LDAPS secure port (`-P`). If you do not specify a port number, `dsconf` uses the default port (`389` for LDAP). You must have LDAP access permissions to configuration data, for example, as the user cn=admin,cn=Administrators,cn=config.

Obtaining Help for Using `dsadm`, `dsconf`, and `dsutil`

For complete information about how to use the `dsadm`, `dsconf`, and `dsutil` commands, see the `dsadm(1M)`, `dsconf(1M)`, and `dsutil(1M)` man pages.

- To obtain a list of subcommands, type the appropriate command:

  ```
  $ dsadm --help
  ```

  ```
  $ dsconf --help
  ```

  ```
  $ dsutil --help
  ```

- To obtain information about how to use a subcommand, type the appropriate command:

  ```
  $ dsadm subcommand --help
  ```

  ```
  $ dsconf subcommand --help
  ```

  ```
  $ dsutil subcommand --help
  ```

Modifying Configuration Properties by Using `dsconf`

Many of the `dsconf` subcommands enable you to view and modify configuration properties.

- To list the configuration properties used in Directory Server, type:

  ```
  $ dsconf help-properties
  ```

- To find a particular property, search the output of the help properties.

  For example, if you are using a UNIX® platform and you want to search for all properties relating to referrals, use the following command.

  ```
  $ dsconf help-properties | grep -i referral
  SER  referral-url                      rw  M  LDAP_URL | undefined
       Referrals returned to clients requesting a DN not stored in this
       Directory Server (Default: undefined)
  SUF  referral-mode                     rw     disabled|enabled|only-on-write
       Specifies how referrals are used for requests involving the suffix
       (Default: disabled)
  SUF  referral-url                      rw  M  LDAP_URL | undefined
       Server(s) to which updates are referred (Default: undefined)
  SUF  repl-rewrite-referrals-enabled    rw     on|off
       Specifies whether automatic referrals are overwritten (Default: off)
  ```

  Note that the properties are grouped by targeted objects, such as suffixes (SUF) and server (SER). The `rw` keyword indicates that the property is readable and writable. The `M` keyword indicates that the property is multivalued.

- To see the server attribute, use verbose mode. For example, on a UNIX system, type:

```
$ dsconf help-properties -v | grep -i referral-mode
SUF  referral-mode    rw  disabled|enabled|only-on-write  nsslapd-state
  Specifies how referrals are used for requests involving the suffix
  (Default: disabled)
```

For more information about individual properties, see the man page for that property. The man pages are in *Sun Directory Server Enterprise Edition 7.0 Man Page Reference*.

Setting Multi-Valued Properties With `dsconf`

Certain Directory Server properties can take multiple values. The syntax to specify these values is as follows:

```
$ dsconf set-container-prop -h host -p port container-name \
 property:value1 property:value2
```

For example, to set multiple encryption ciphers for a server, use the following command:

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
```

To add a value to a multi-valued property that already contains values, use the following syntax:

```
$ dsconf set-container-prop -h host -p port container-name property+:value
```

To remove a value from a multi-valued property that already contains values, use the following syntax:

```
$ dsconf set-container-prop -h host -p port container-name property-:value
```

For example, in the scenario described previously, to add the `SHA` encryption cipher to the list of ciphers, run this command:

```
$ dsconf set-server-prop -h host1 -p 1389 \
 ssl-cipher-family+:TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
```

To remove the `MD5` cipher from the list, run this command:

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family-:SSL_RSA_WITH_RC4_128_MD5
```

Working With the `dsutil` Command

You must create the following ACIs to work with the `dsutil` command successfully:

```
$ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w - -c
dn: cn=config
changetype: modify
```

```
add: aci
aci: (targetattr="*")(version 3.0; acl "Allow the Suffix Manager to browse the tree"; \
allow (read,search,compare)userdn = "ldap:///$USERSFXADMIN";)
aci: (targetattr="nsslapd-rootpw")\
(version 3.0; acl "Prevent the Suffix Manager from accessing passwords"; \
deny (all)userdn = "ldap:///$USERSFXADMIN";)
aci: (targetattr="userPassword")\
(version 3.0; acl "Prevent the Suffix Manager from accessing passwords"; \
deny (all)userdn = "ldap:///$USERSFXADMIN";)
aci: (targetattr="dsKeyedPassword")\
(version 3.0; acl "Prevent the Suffix Manager from accessing passwords"; \
deny (all)userdn = "ldap:///$USERSFXADMIN";)
```

For more information about `dsutil` command, see `dsutil(1M)`.

Man Pages

The man pages provide descriptions of all commands and attributes used in Directory Server. In addition, the man pages show some useful examples of how to use the commands in deployment.

◆ ◆ ◆ CHAPTER 2

Directory Server Instances and Suffixes

This chapter describes how to create and manage Directory Server instances and suffixes. Many other directory administration tasks are configured at the suffix level, but they are covered in other chapters in this book.

This chapter covers the following topics:

Quick Procedure for Creating Server Instances and Suffixes

This chapter contains detailed information about how to create server instances and suffixes. If you need to quickly create a Directory Server instance and suffix, and import some example data, see "Checking Your Directory Server Enterprise Edition Installation" in *Sun Directory Server Enterprise Edition 7.0 Installation Guide*.

Creating and Deleting a Directory Server Instance

This section shows how to create and delete a Directory Server instance.

▼ To Create a Directory Server Instance

Before you can administer data, you must create a Directory Server instance by using command-line tools or the browser interface Directory Service Control Center (DSCC). In DSCC, a Directory Server instance is often referred to simply as a "Directory Server".

When you create a Directory Server instance, the files and directories required for your Directory Server are created in the *instance-path* that you specify.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

If you use DSCC to create a new server instance, you can choose to copy some or all server configuration settings from an existing server.

The `dsadm` command enables you to manage a Directory Server instance and the files belonging to that instance on the local host. The command does not let you administer servers over the network, but only directly on the local host. The `dsadm` command has subcommands for each key management task. For a complete description, see `dsadm`(1M).

The `dsconf` command is an LDAP client. The command enables you to configure nearly all server settings on a running Directory Server instance from the command line. You can configure settings whether the server is on the local host or another host that is accessible across the network. The `dsconf` command has subcommands for each key configuration task. For a complete description, see `dsconf`(1M).

1 **Create a new Directory Server instance and set the instance path.**

```
$ dsadm create instance-path
```

You are prompted to set a password for the Directory Manager for this server.

To specify a non-default port number for the server instance, or any other parameter, see the `dsadm`(1M) man page.

For example, to create a new instance in the directory `/local/dsInst`, use this command:

```
$ dsadm create /local/dsInst
Choose the Directory Manager password:
Confirm the Directory Manager password:
Use 'dsadm start /local/dsInst' to start the instance
```

The instance is created in a directory on the local file system and not a network file system.

2 Check that the server instance has been created correctly.

```
$ dsadm info instance-path
```

For example:

```
$ dsadm info /local/dsInst
Instance Path:         /local/dsInst
Owner:                 user1(group1)
Non-secure port:       1389
Secure port:           1636
Bit format:            64-bit
State:                 Running
Server PID:            22555
DSCC url:              -
SMF application name:  -
Instance version:      D-A00
```

3 (Optional) If you installed Directory Server using the native packages and your operating system provides a service management solution, you can enable the server to be managed as a service, as shown in this table.

Operating System	Command
Solaris 10	`dsadm enable-service --type SMF` *instance-path*
Solaris 9	`dsadm autostart` *instance-path*
Windows	`dsadm enable-service --type WIN_SERVICE` *instance-path*

4 Start Directory Server.

```
$ dsadm start instance-path
```

Note – The server is running but does not contain data or a suffix. Use `dsconf` to create a suffix.

---

5 (Optional) Register the server instance with Directory Service Control Center by using either of the following methods.

- **Login to DSCC, and then use the Register Existing Server action on the Servers tab of the Directory Servers tab.**

  Access DSCC using `http://hostname:8080/dscc7` or `https://hostname:8181/dscc7` as per your application server configuration.

- **Use the command** `dsccreg add-server`**.**

  ```
  $ dsccreg add-server -h hostname --description "My DS" /local/dsInst
  Enter DSCC administrator's password:
  ```

```
/local/dsInst is an instance of DS
Enter password of "cn=Directory Manager" for /local/dsInst:
This operation will restart /local/dsInst.
Do you want to continue ? (y/n) y
Connecting to /local/dsInst
Enabling DSCC access to /local/dsInst
Restarting /local/dsInst
Registering /local/dsInst in DSCC on hostname.
```

See `dsccreg(1M)` for more information about the command.

6 **If you want to use a password policy and your Directory Server instance is standalone, or if it belongs to a replication topology that has already been migrated to `to-DS6-only` password policy mode, move the instance to that mode.**

```
$ dsconf pwd-compat -h host -p port to-DS6-migration-mode

## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
$ dsconf pwd-compat -h host -p port to-DS6-mode

## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
```

The above action should be performed in the specified sequence.

For more information about password policies compatibilities, see "Password Policy" in *Sun Directory Server Enterprise Edition 7.0 Upgrade and Migration Guide*.

▼ To Delete a Directory Server Instance

Before you delete the server instance, you must prepare the instance for deletion. Refer to the following procedure to delete a server instance successfully:

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Stop the Directory Server.**

```
$ dsadm stop [--force] instance-path
```

2 **If the server was previously registered with DSCC, unregister the server using the following command:**

```
$ dsccreg remove-server /local/dsInst
Enter DSCC administrator's password:
/local/dsInst is an instance of DS
Enter password of "cn=Directory Manager" for /local/dsInst:
This operation will restart /local/dsInst.
Do you want to continue ? (y/n) y
Unregistering /local/dsInst from DSCC on localhost.
Connecting to /local/dsInst
Disabling DSCC access to /local/dsInst
Restarting /local/dsInst
```

For details, see the `dsccreg(1M)` man page.

3 **(Optional) If you previously enabled the server instance in a service management solution, disable the server from being managed as a service.**

Operating System	Command
Solaris 10	`dsadm disable-service --type SMF` *instance-path*
Solaris 9	`dsadm autostart --off` *instance-path*
Windows	`dsadm disable-service --type WIN_SERVICE` *instance-path*

4 **Delete the server instance.**

```
$ dsadm delete instance-path
```

Caution – This command removes everything, under the *instance-path* directory.

If the instance has been enabled as a service, or if the instance is started automatically at system startup, `dsadm delete` requires root access.

Starting, Stopping, and Restarting a Directory Server Instance

To start, stop or restart, the server from the command line, use the commands `dsadm start`, `dsadm stop`, and `dsadm restart`, respectively.

Note – When you stop and restart a Directory Server instance with a large cache in memory configured to hold entries, the cache takes some time to refill. While the cache fills again, the instance responds more slowly.

These commands must be run by the same UID and GID that created the Directory Server, or run by root. For example, if Directory Server runs as `user1` , you should run the `start`, `stop`, and `restart` utilities as `user1`.

Note – On Solaris, role-based access control allows you to run Directory Server as a user other than root.

▼ To Start, Stop, and Restart Directory Server

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. However, this does not apply to the step for enabling and disabling service management. Enabling and disabling service management must be done at the command line when starting and stopping Directory Server.

For more information about `dsadm` subcommands and options used below, see `dsadm(1M)`.

- **To start, stop, or restart Directory Server, do one of the following:**
  - To start the server, type:

    ```
    $ dsadm start instance-path
    ```

    For example, to start a server with the instance path `/local/dsInst`, use this command:

    ```
    $ dsadm start /local/dsInst
    ```

    If the start operation fails after a configuration change, use the `--safe` option as shown in the following command:

    ```
    $ dsadm start --safe /local/dsInst
    ```
  - To stop the server, type:

    ```
    $ dsadm stop [--force] instance-path
    ```

    For example:

    ```
    $ dsadm stop --force /local/dsInst
    ```
  - To restart the server, type:

    ```
    $ dsadm restart instance-path
    ```

    For example:

```
$ dsadm restart /local/dsInst
```

▼ To List All the Running Instances

- **List the running instances on a host using the following command:**

  ```
  dsadm list-running-instances [--all]
  ```

  The –`all` option lists the running instances from any installation path.

▼ To Stop the Running Instances

- **Stop the running instances on a host using the following command:**

  ```
  dsadm stop-running-instances [-i] [--force]
  ```

Creating Suffixes

After you have created your Directory Server instance, you must create one or more suffixes for the server's Directory Information Tree (DIT). The DIT consists of all of the entries in your server, as identified by their distinguished names (DNs). The hierarchical nature of a DN creates branches and leaves that structure the data in the tree. The DIT is defined and managed administratively in terms of suffixes and sub-suffixes. DSCC provides controls for creating and administering all of these elements. Alternatively, you can use command-line tools.

For conceptual information about structuring directory data and about suffixes in general, refer to the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

As explained in the following procedure, you can use the `dsconf create-suffix` command to create a suffix configuration in your directory. Because root suffixes and sub-suffixes are managed internally in the same way, the procedure for creating them from the command line is nearly the same. The procedure shows the `dsconf create-suffix` command used only with the required options. For more information about other options of this command, see the `dsconf`(1M) man page or run the following command:

```
$ dsconf create-suffix --help
```

The configuration entries can be created by any administration user. However, the top entry of the suffix *must* be created by the Directory Manager or as a Directory Administrator, such as `cn=admin,cn=Administrators,cn=config`.

▼ To Create a Suffix

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

If you use DSCC to create a new suffix, you can choose to copy some or all suffix configuration settings from an existing suffix.

1 Create the root suffix.

Ensure that your server is running, then type this command:

```
$ dsconf create-suffix -h host -p port suffix-DN
```

where the *suffix-DN* is the full DN of the new suffix. For a root suffix, the convention is to use the domain-component (`dc`) naming attribute.

For example, to create a suffix for the DN `dc=example,dc=com` , use this command:

```
$ dsconf create-suffix -h host1 -p 1389 dc=example,dc=com
```

This command creates the new suffix as follows:

- The top level (or base) entry of the root suffix is created.
- The configuration entries in `cn=config` for both the suffix and the database are created.
- The default database name is based on the suffix DN.

For information about all of the suffixes, including the new suffix that has been created, use this command:

```
$ dsconf list-suffixes -h host -p port -v
```

The `-v` option displays verbose mode, which shows how many entries are on the suffix, and any replication information.

Note – If you have more than one Directory Server instance, use the `-h` *host name* and `-p` *port number* options to specify which server instance the suffix should belong to.

If you want to specify a non-default path for the database files, use the `-L` option. You can change the suffix database path at a later stage. To do this, use the command `dsconf set-suffix-prop` *suffix-DN* `db-path:` *new-db-path*, then stop the server, move the database files manually, and restart the server.

To see all the options that you can use when creating suffixes, refer to the `dsconf(1M)` man page.

Note – Database names can contain only ASCII (7-bit) alphanumeric characters, hyphens (-), and underscores (_). Directory Server does not accept multibyte characters (such as in Chinese or Japanese character sets) in strings for database names, file names, and path names.

To work around this issue, when creating a Directory Server suffix having multibyte characters, specify a database name that has no multibyte characters. When creating a suffix on the command line, for example, explicitly set the `--db-name` option of the `dsconf create-suffix` command.

```
$ dsconf create-suffix --db-name asciiDBName UTF-8SuffixDN
```

Do not use `default` as database name for the suffix. Do not use multibyte characters for the database name.

2 **If required, create the sub-suffix:**

```
$ dsconf create-suffix -h host -p port subSuffix-DN
```

then attach the sub-suffix to the root suffix.

```
$ dsconf set-suffix-prop -h host -p port subSuffix-DN parent-suffix-dn:parentSuffix-DN
```

where *parentSuffix-DN* must have the same value as *suffix-DN* in the previous step. The *suffix-DN* for the sub-suffix includes the relative distinguished name (RDN) of the sub-suffix and the DN of its parent suffix.

For example, to create the sub-suffix `ou=Contractors,dc=example,dc=com`, and to attach the sub-suffix to the root suffix, type:

```
$ dsconf create-suffix -h host1 -p 1389 ou=Contractors,dc=example,dc=com
$ dsconf set-suffix-prop -h host1 -p 1389 ou=Contractors,dc=example,dc=com \
 parent-suffix-dn:dc=example,dc=com
```

When this entry is added to the directory, the database module of the server automatically creates the database files in the following directory:

instance-path`/db/`*database-name*

where *database-name* is the name automatically built from a part of the suffix. For example, in the previous example, the *database-name* would be `Contractors`

3 **(Optional) Initialize the suffix with data. See "Initializing a Suffix" on page 52.**

Disabling or Enabling a Suffix

Sometimes, you might need to make a suffix unavailable for maintenance, or to make its contents unavailable for security reasons. The action of disabling a suffix prevents the server from reading or writing the contents of the suffix in response to any client operations. When you disable a suffix, you no longer have access to that suffix, and the referral mode is automatically set to disabled.

▼ To Disable then Enable a Suffix

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Disable the suffix.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:off
```

Note – You cannot disable a suffix on which replication is enabled because most properties of a replicated suffix are determined by the replication mechanism.

---

2 **Enable the suffix.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:on
```

Setting Referrals and Making a Suffix Read-Only

If you want to limit access to a suffix without disabling the suffix completely, you can modify the access permissions to allow read-only access. In this case you must define a referral to another server for write operations. You can also deny both read and write access, and define a referral for all operations on the suffix.

Referrals can also be used to temporarily point a client application to a different server. For example, while backing up the contents of the suffix, you might add a referral to another suffix.

If your suffix is a consumer in a replicated environment, the replication mechanism determines the value of the referral setting. Although you can manually modify the referral setting, the referral will be overwritten at the next replication update. For information about setting replication referrals, see "To Perform Advanced Consumer Configuration" on page 257.

Referrals are labeled URLs, that is, an LDAP URL optionally followed by a space character and a label. For example:

```
ldap://phonebook.example.com:389/
```

Or:

```
ldap://phonebook.example.com:389/ou=All%20People,dc=example,dc=com
```

Because space characters are significant, any space characters in the URL part of the referral must be escaped using `%20`.

▼ To Set Referrals to Make a Suffix Read-Only

1. **Set the referral URL.**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:LDAP-URL
   ```

   where *LDAP-URL* is a valid URL containing the host name, port number, and DN of the target.

   For example:

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
    referral-url:ldap://phonebook.example.com:389/
   ```

   You can specify any number of LDAP URLs.

2. **Set the referral mode in order to make the suffix read-only.**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:only-on-write
   ```

   To make the suffix unavailable for both read and write operations, and to return referrals for all requests, set the `referral-mode` to `enabled`.

3. **As soon as the command is successful, the suffix is read-only or inaccessible and ready to return referrals.**

4. **(Optional) When the suffix becomes available, disable the referrals to make the suffix read-write again.**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:disabled
   ```

   When referrals are disabled, the suffix automatically becomes read-write, unless you have disabled the suffix itself by setting the `enabled` property of the suffix to `off`.

Importing Data From an LDIF File

You can import data to a Directory Server suffix in the following ways:

- Initialize a suffix from an LDIF file. This operation deletes the current data in the suffix and replaces it with the contents of the LDIF file.
- Use an LDIF file to perform bulk `ldapadd`, `ldapmodify`, or `ldapdelete` operations. This allows you to add, modify, and delete entries in bulk in any suffix of the directory.

Note – The offline import (`dsadm import`) does not remove the changelog as the changelog data may still be in the suffix. At server start, replication decides if the changelog needs to be kept or not. Online import (`dsconf import`) decides straight away if changelog needs to be recreated or not.

The following table shows the differences between initializing a suffix and adding, modifying, and deleting entries in bulk.

TABLE 2–1 Comparison of Initializing a Suffix and Importing Data in Bulk

Domain of Comparison	Initializing Suffixes	Adding, Modifying, and Deleting Entries in Bulk
Content	Overwrites content	Does not overwrite content
LDAP operations	N/A	Add, modify, delete
Performance	Fast	Slower
Response to server failure	Atomic (all changes are lost after a failure)	Best effort (all changes made up to the point of the failure remain)
LDIF file location	Accessible from server	On client machine

Commands	If server is local and stopped:
	`dsadm import`
	If server is remote and running:
	`dsconf import`
		`ldapmodify -B`
		Note – Bulk import using the `ldapmodify -B` command erases the existing entries under the target suffix.

Initializing a Suffix

Initializing a suffix overwrites the existing data in a suffix with the contents of an LDIF file that contains only entries for addition.

You must be authenticated as the Directory Manager or an Administrator to initialize a suffix.

When the server is running, only the Directory Manager and Administrators can import an LDIF file that contains a root entry. For security reasons, only these users have access to the root entry of a suffix, for example, `dc=example,dc=com`.

Before restoring suffixes involved in replication agreements, read "Restoring Replicated Suffixes" on page 227.

▼ To Initialize a Suffix

Note – All LDIF files that you import must use UTF-8 character-set encoding.

When initializing a suffix, the LDIF file must contain the root entry and all directory tree nodes of the corresponding suffix.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Use one of the following commands to initialize the suffix from an LDIF file, that is, import the contents of a database to an LDIF file.**

Caution – These commands overwrite the data in your suffix.

- If your server is local and stopped, type:

  ```
  $ dsadm import instance-path LDIF-file suffix-DN
  ```

  The following example uses the `dsadm import` command to import two LDIF files into a single suffix:

  ```
  $ dsadm import /local/dsInst /local/file/example/demo1.ldif \
   /local/file/example/demo2.ldif dc=example,dc=com
  ```

- If your server is running (local or remote), type:

  ```
  $ dsconf import -h host -p port LDIF-file suffix-DN
  ```

  The following example imports an LDIF file using `dsconf import`. You do not need root privileges to run the command, but you must authenticate as a user with root permissions, such as the Directory Manager.

  ```
  $ dsconf import -h host1 -p 1389 /local/file/example/demo1.ldif \
   ou=People,dc=example,dc=com
  ```

Note – When using the `dsconf` command, the `dse.ldif` file must be available to the host running the server.

For more information, see the `dsadm`(1M) and `dsconf`(1M)man pages.

▼ To Load Sample Data in Directory Server Instance

Examples that use command-line tools depend on sample data residing under the `dc=example,dc=com` suffix of your directory.

You can set up part of the data that is required by creating a `dc=example,dc=com` suffix. You can then populate the suffix with entries from the *install-path*`/dsee7/resources/ldif/Example.ldif` file.

1. **Create a new Directory Server instance and start the instance.**

   ```
   $ dsadm create -p port -P SSL-port instance-path
   $ dsadm start instance-path
   ```

2. **Read the `Example.ldif` file to find bind passwords needed in the examples.**

3. **Create suffix and load the `Example.ldif` content into the directory by using the following commands:**

   ```
   $ dsconf create-suffix -h localhost -p 1389 dc=example,dc=com
   $ dsconf import -h localhost -p 1389 \
   install-path/dsee7/resources/ldif/Example.ldif dc=example,dc=com
   ```

   For more information, see "To Create a Directory Server Instance" on page 42.

4. **Generate test data for examples by using the `makeldif`(1) command, as shown in the next step, and the following template:**

   ```
   define suffix=dc=example,dc=com
   define maildomain=example.com

   branch: ou=test,[suffix]
   subordinateTemplate: person:100

   template: person
   rdnAttr: uid
   objectclass: top
   objectclass: person
   objectclass: organizationalPerson
   objectclass: inetOrgPerson
   givenName: <first>
   sn: <last>
   ```

```
cn: {givenName} {sn}
initials: {givenName:1}{sn:1}
employeeNumber: <sequential>
uid: test{employeeNumber}
mail: {uid}@[maildomain]
userPassword: auth{employeeNumber}{employeeNumber}
telephoneNumber: <random>
description: This is the description for {cn}.
```

5 **Create a `test.template` file and copy the template content, as shown above, into it. Use commands such as the following to generate the data in `test.ldif` and to load the content into the directory.**

Note – The `test.template` file must be created in the *install-path*`/dsee7/dsrk/bin/example_files` directory.

```
$ cd install-path/dsee7/dsrk/bin/example_files
$ ../makeldif -t test.template -o test.ldif
Processing complete.
101 total entries written.
$ ../ldapmodify -a -c -D uid=hmiller,dc=example,dc=com -w - -f test.ldif
Enter bind password:
|PO
```

If you read `Example.ldif`, you see that the password for `hmiller` is `hillock`.

Note – This step is specific to the zip installation because the `makeldif` command is available only in the zip distribution.

Adding, Modifying, and Deleting Entries in Bulk

When you perform an `ldapmodify` operation, you are able to add, modify, or delete entries in bulk. Entries are specified in an LDIF file that contains update statements to modify or delete existing entries. This operation does not erase entries that already exist.

The changed entries may target any suffix that is managed by your Directory Server. As with any other operation that adds entries, the server will index all new entries as they are imported.

The `ldapmodify` command will import an LDIF file through LDAP and perform all operations that the file contains. Using this command you can modify data in all directory suffixes at the same time.

Before restoring suffixes involved in replication agreements, see "Restoring Replicated Suffixes" on page 227.

▼ To Add, Modify and Delete Entries in Bulk

Note – All LDIF files that you import must use UTF-8 character-set encoding.

When importing an LDIF file, parent entries must either exist in the directory or be added first from the file.

- **Add, modify, or delete from an LDIF file in bulk.**

  `$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -B` *baseDN* `-f` *LDIF-file*

  The following example performs an import using the `ldapmodify` command. You do not need root privileges to run this command, but you must authenticate as a user with root permissions, such as `cn=Directory Manager` or `cn=admin,cn=Administrators,cn=config`. The last parameter specifies the name of the LDIF file to import.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - \
 -B dc=example,dc=com -f /local/dsInst/ldif/demo.ldif
```

Deleting a Suffix

Deleting a suffix removes its entire branch from the DIT.

Note – When you delete a suffix, you permanently remove all of its data entries from the directory. You also remove all suffix configuration information, including its replication configuration.

You cannot delete a parent suffix and keep its sub-suffixes in the DIT as new root suffixes. If you want to delete an entire branch that contains sub-suffixes, you must also delete the sub-suffixes of the deleted parent and their possible sub-suffixes.

▼ To Delete a Suffix

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Remove the suffix configuration entry:**

  `$ dsconf delete-suffix -h` *host* `-p` *port* `[`*subSuffix-DN*`]` *suffix-DN*

  This command removes the suffix from the server, starting with the base entry at the *suffix-DN*. The suffix is no longer visible or accessible in the directory.

Compacting a Suffix

Directory Server 7.0 supports the compaction of suffixes offline. Online compaction is not supported in this release. If storage space is available, compacting a suffix reduces the size of the database by reorganizing the database keys, and returns the storage space to the filesystem, if possible. Since application data is not processed and based on the way the database keys are managed, it is not guaranteed that the storage space can be reclaimed. Nevertheless database size will not grow.

▼ To Compact a Suffix Offline

Stop the server and backup your database before performing this task.

- **Compact the required suffix.**

  ```
  $ dsadm repack instance-path suffix-dn
  ```

  All `.db3` files related to the specified suffix are compacted.

  If you run this command with the `-b` option, you can specify a back end database name, instead of a suffix DN. At least one suffix, or one back end must be specified.

◆ ◆ ◆ CHAPTER 3

Directory Server Configuration

This chapter describes how to configure Directory Server. You can use the `dsconf` command (see the `dsconf(1M)` man page).

You can also use Directory Service Control Center (DSCC), which is the preferred method. DSCC makes additional checks during the configuration process, which can minimize errors. In addition, DSCC enables you to copy the configuration of one server instance to another server instance. For more information about using DSCC, see the DSCC online help.

Displaying the Configuration of Directory Server Instance

To display the configuration of Directory Server instance, run `dsconf info`.

```
$ dsconf info -h host -p port
Instance path   :  instance path
Global State    :  read-write
Host Name       :  host
Port            :  port
Secure port     :  secure port
Total entries   :  20844

Suffixes        :  suffix-DN

Dest. Servers   :  host:port

On-Going Tasks  :  import
Finished Tasks  :  backup

==================
Suffix suffix-DN

Replication role            :  master
```

```
Attribute to reindex        :  mail
                               country
                               telephonenumber
Index filter analyzer       :  Running for suffix-DN (Nov 7, 2008 4:56:33 PM)
```

The above output assumes that you have created suffixes, and replication agreements with the destination servers. It also displays the ongoing import operation and finished backup operation.

The above output also displays the suffix related information such as Replication role, Attribute to reindex, and Index filter analyzer. If no specific information is available, no such information is displayed in `dsconf info` output.

Modifying the Configuration Using DSCC

The recommended method for modifying the configuration is to use DSCC. This browser interface provides task-based controls to help you set up your configuration quickly and efficiently. Using DSCC, you can modify a configuration setting on one server and then copy that configuration setting to other servers. In addition, the DSCC interface manages the complexity and interdependence of the configuration for you. Detailed procedures for modifying the configuration using DSCC can be found in the DSCC online help.

Modifying the Configuration From the Command Line

You can automate configuration tasks by writing scripts that use command-line tools.

Modify the configuration through the command line by using the `dsconf` command. This command uses LDAP to modify the `cn=config` subtree. For more information about `dsconf`, see "Directory Server Command-Line Tools" on page 35.

For any tasks that you cannot perform using `dsconf`, use the `ldapmodify` command.

Note – If you want to modify the server configuration properties by using the command `dsconf set-server-prop`, you need to know which properties you can modify and their default values. Use this command to display help on all properties:

```
$ dsconf help-properties -v
```

Search the property help for the item that you need. For example, on a UNIX platform, type the following to search for memory cache properties:

```
$ dsconf help-properties -v | grep cache
```

For more information about configuration entries in `cn=config` and for a complete description of all configuration entries and attributes, including the range of allowed values, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

Modifying the `dse.ldif` File

Directory Server stores all of its configuration information in this file:

instance-path`/config/dse.ldif`

Caution – Modifying the configuration by editing the contents of the `dse.ldif` file directly is prone to error and is not recommended. However, if you choose to edit this file manually, stop the server before you edit the file and restart it after you have finished editing.

The `dse.ldif` file is in the LDAP Data Interchange Format (LDIF). LDIF is a textual representation of entries, attributes, and their values, and is a standard format described in RFC 2849 (`http://www.ietf.org/rfc/rfc2849`).

The Directory Server configuration in the `dse.ldif` file consists of the following:

- The attributes and values of the `cn=config` entry.
- All of the entries in the subtree below `cn=config` and their attributes and values.
- The object classes and access control instructions of the root entry ("") and the `cn=monitor` entry. The other attributes of these entries are generated by the server.

  Only the system user who owns the Directory Server instance has the rights to read and write the file.

  Directory Server makes most configuration settings readable and writable through LDAP. By default, the `cn=config` branch of the directory can be read by anyone with authorization and can be written to only by the Directory Manager (`cn=Directory Manager`) and to the

administrative users under cn=Administrators,cn=config. The administration user can view and modify the configuration entries just like any other directory entry.

Do not create non-configuration entries under the cn=config entry because they will be stored in the dse.ldif file, which is not the same highly scalable database as regular entries. As a result, if many entries, and particularly entries that are likely to be updated frequently, are stored under cn=config, performance will likely be degraded. However, it can be useful to store special user entries such as the Replication Manager (supplier bind DN) entry under cn=config, to centralize configuration information.

Note – If dse.ldif refers to a nonexistent Sun Microsystems plug-in, it can be considered to have a valid signature. The following warning message is displayed:

```
WARNING<4227> - Plugins - conn=-1 op=-1 msgId=-1 -
Detected plugin paths from another install, using current install.
```

This warning message appears only for plug-ins with a vendor of Sun Microsystems.

Configuring Administration Users

Directory Server contains default administration users, the Directory Manager and the cn=admin,cn=Administrators,cn=config user. Both of these users have the same access rights, but cn=admin,cn=Administrators,cn=config is subject to ACIs.

This section explains how to create an administration user with root access, and how to configure the Directory Manager.

▼ To Create an Administration User with Root Access

If you want to create a new administration user with the same rights as cn=admin,cn=Administrators,cn=config, create the new user in the group cn=Administrators,cn=config. All users in this group are subject to a global ACI that allows the same access as the Directory Manager.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Create a new administration user.**

  For example, to create a new user cn=Admin24,cn=Administrators,cn=config, type:

  ```
  $ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
  dn: cn=admin24,cn=Administrators,cn=config
  changetype: add
  ```

```
objectclass: top
objectclass: person
userPassword: password
description: Administration user with the same access rights as Directory Manager.
```

The `-D` and `-w` options give the bind DN and password, respectively, of a user with permissions to create this entry.

▼ To Configure the Directory Manager

The Directory Manager is the privileged server administrator, comparable to the `root` user on UNIX systems. Access control does not apply to the Directory Manager.

For most administration tasks, you are not required to use the Directory Manager. Instead, you can use the user `cn=admin,cn=Administrators,cn=config`, or any other user that you create beneath `cn=Administrators,cn=config`. The only tasks that require the Directory Manager are changing the root ACI, and replication troubleshooting tasks, such as repairing replication and searching tombstones.

You can change the Directory Manager DN and password, as well as create a file from which the password can be automatically read.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Find the existing Directory Manager DN.

```
$ dsconf get-server-prop -h host -p port root-dn
root-dn:cn=Directory Manager
```

2 Modify the Directory Manager settings as required.

- To modify the Directory Manager DN, type:

  ```
  $ dsconf set-server-prop -h host -p port root-dn:new-root-dn
  ```

  Use quotes if there are spaces in the Directory Manager DN. For example:

  ```
  $ dsconf set-server-prop -h host1 -p 1389 root-dn:"cn=New Directory Manager"
  ```

- To change the Directory Manager password, type:

  Create a temporary file for setting the password. This file is read once, and the password is stored for future use.

  ```
  $ echo password > /tmp/pwd.txt
  ```

  Set the server root password file property.

  ```
  $ dsconf set-server-prop -h host -p port root-pwd-file:/tmp/pwd.txt
  ```

This command prompts the server to read the password file. Remove the temporary password file after you have set the password file property.

```
$ rm /tmp/pwd.txt
```

Protecting Configuration Information

The root Directory Server entry (the entry returned for a base object search with a zero-length DN "") and the subtrees below `cn=config`, `cn=monitor`, and `cn=schema` contain access control instructions (ACIs) that are automatically generated by Directory Server. These ACIs are used to determine user permissions to directory entries. These ACIs are sufficient for evaluation purposes. However, for any production deployment, you need to evaluate your access control requirements and design your own access controls.

If you want to hide the existence of one or more additional subtrees and protect your configuration information for security reasons, you must place additional ACIs on the DIT.

- Place an ACI attribute in the entry at the base of the subtree you want to hide.
- Place an ACI in the root DSE entry on the `namingContexts` attribute. The root DSE entry attribute called `namingContexts` contains a list of the base DNs for each of the Directory Server databases.
- Place an ACI on the `cn=config` and `cn=monitor` subtrees. The subtree DNs are also stored in the mapping tree entries below `cn=config` and `cn=monitor`.

For more information about creating ACIs, see Chapter 6, “Directory Server Access Control.”

Changing Directory Server Port Numbers

You can modify the LDAP port or the LDAPS secure port number of your user directory server by using DSCC or by using the `dsconf set-server-prop` command.

If you change a port number, be aware of the following:

- If you set a non-privileged port number and Directory Server is installed on a machine to which other users have access, you might expose the port to a hijack risk by another application. In other words, another application can bind to the same address/port pair. This rogue application might then be able to process requests intended for Directory Server. That is, the rogue application could be used to capture passwords used in the authentication process, to alter client requests or server responses, or to produce a denial of service attack. To avoid this security risk, use the `listen-address` or `secure-listen-address` properties to specify the interface (address) on which Directory Server listens.

If you change the port number by using the command line, be aware of the following:

- If the Directory Server is referenced in replication agreements that are defined on other servers, the replication agreements must be updated to use the new port number.
- If you have used DSCC previously to manage the server, the server will be temporarily unable to be viewed after the change in port number. To view the server again, you must unregister the server and then register it again in DSCC using the new port number.

▼ To Modify a Port Number, Enable a Port, and Disable a Port

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Note – Once you make your modifications, you must restart the server for the changes to take effect.

---

1 Verify the existing settings for your port.

```
$ dsconf get-server-prop -h host -p port port-type
```

Where *port-type* is one of the following:

`ldap-port`	LDAP default port
`ldap-secure-port`	LDAPS secure port
`dsml-port`	DSML default port
`dsml-secure-port`	DSML secure port

For example, to display the LDAPS secure port, type:

```
$ dsconf get-server-prop -h host1 -p 2501 ldap-secure-port
Enter "cn=Directory Manager" password:
ldap-secure-port : 2511
```

If the returned result is an integer, the port is enabled. If the returned result is `disabled`, the port is disabled.

Note – You can also list the LDAP default port and LDAPS secure port using the `dsadm`

---

2 If required, modify a port number or enable a port.

```
$ dsconf set-server-prop -h host -p port port-type:new-port
```

For example, to change the LDAP port number from 1389 to 1390, use this command:

```
$ dsconf set-server-prop -h host1 -p 1389 ldap-port:1390
```

To enable the DSML secure port on port number 2250, use this command:

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:2250
```

3 **If required, disable a port.**

```
$ dsconf set-server-prop -h host -p port port-type:disabled
```

For example, to disable the DSML secure port, use the command:

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:disabled
```

Configuring DSML

In addition to processing requests in the Lightweight Directory Access Protocol (LDAP), Directory Server also responds to requests sent in the Directory Service Markup Language version 2 (DSMLv2). DSML is another way for a client to encode directory operations. The server processes DSML as any other request, with all of the same access control and security features. DSML processing allows many other types of clients to access your directory contents.

Directory Server supports the use of DSMLv2 over the Hypertext Transfer Protocol (HTTP/1.1) and uses the Simple Object Access Protocol (SOAP) version 1.1 as a programming protocol to transport the DSML content. For more information about these protocols and for examples of DSML requests, see Chapter 13, "Directory Server DSMLv2," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This section covers the following topics:

▼ To Enable the DSML-over-HTTP Service

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Set the DSML mode to** `on`**.**

```
$ dsconf set-server-prop -h host -p port dsml-enabled:on
```

2 **Set the secure DSML port.**

```
$ dsconf set-server-prop -h host -p port dsml-secure-port:port
```

3 **Set the non—secure DSML port.**

```
$ dsconf set-server-prop -h host -p port dsml-port:port
```

By default, this port is set to `disabled`

4 **Restart the server.**

```
$ dsadm restart instance-path
```

Next Steps According to the parameters and attribute values you defined, DSML clients may use the following URLs to send requests to this server:

`http://`*host*`:`*DSML-port*`/`*relative-URL*

`https://`*host*`:`*secure-DSML-port*`/`*relative-URL*

Note – The *relative-URL* can be read and set using the `dsml-relative-root-url` property.

▼ To Disable the DSML-over-HTTP Service

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Set the DSML mode to `off`.**

```
$ dsconf set-server-prop -h host -p port dsml-enabled:off
```

2 **Set the secure DSML port to `disabled`.**

```
$ dsconf set-server-prop -h host -p port dsml-secure-port:disabled
```

3 **Restart the server.**

```
$ dsasm restart instance-path
```

▼ To Configure DSML Security

You can configure the level of security that is required to accept DSML requests. To do this, you must configure DSML client authentication.

- **Set the DSML client authentication mode.**

  ```
  $ dsconf set-server-prop -h host -p port dsml-client-auth-mode:dsml-mode
  ```

  By default the `dsml-client-auth-mode` property is set to `client-cert-first`.

  dsml-mode can be one of:

  - `http-basic-only` - This is the default value. The server uses the contents of the HTTP Authorization header to find a user name that can be mapped to an entry in the directory. This process and its configuration are encrypted through SSL but do not use client certification. This is described in "DSML Identity Mapping" on page 68.
  - `client-cert-only` - The server uses credentials from the client certificate to identify the client. With this value, all DSML clients must use the secure HTTPS port to send DSML requests and provide a certificate. The server checks that the client certificate matches an entry in the directory. See Chapter 5, "Directory Server Security," for more information.
  - `client-cert-first` - The server will attempt to authenticate clients first with a client certificate if one is provided. Otherwise, the server will authenticate clients using the contents of the Authorization header.

  If no certificate and no Authorization header is provided in the HTTP request, the server performs the DSML request with anonymous binding. Anonymous binding is also used in the following cases:

  - The client provides a valid Authorization header but no certificate when `client-cert-only` is specified.
  - The client provides a valid certificate but no Authorization header when `http-basic-only` is specified.

  Regardless of the client authentication method, if a certificate is provided but it cannot be matched to an entry, or if the HTTP Authorization header is specified but cannot be mapped to a user entry, the DSML request is rejected with error message 403: "Forbidden".

DSML Identity Mapping

When performing basic authentication without a certificate, Directory Server uses a mechanism called *identity mapping* to determine the bind DN to use when accepting DSML requests. This mechanism extracts information from the Authorization header of the HTTP request to determine the identity to use for binding.

The default identity mapping for DSML/HTTP is given by the following entry in your server configuration.

```
dn: cn=default,cn=HTTP-BASIC,cn=identity mapping,cn=config
objectClass: top
```

```
objectClass: nsContainer
objectClass: dsIdentityMapping
cn: default
dsSearchBaseDN: ou=people
dsSearchFilter: (uid=${Authorization})
```

This configuration indicates that the server should use the HTTP user ID as the `uid` value for a DN stored in a Directory Server suffix. For example, if the HTTP user is `bjensen`, the server tries to execute the bind using the DN `uid=bjensen,ou=people`.

For the mapping to work properly you must therefore complete the value of `dsSearchBaseDN`. For example, you can change the value of `dsSearchBaseDN` to `ou=people,dc=example,dc=com`. Then if the HTTP user is `bjensen`, the server tries to execute the bind using the DN `uid=bjensen,ou=people,dc=example,dc=com`.

```
dn: cn=default,cn=HTTP-BASIC,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
cn: default
dsSearchBaseDN: ou=people,dc=example,dc=com
dsSearchFilter: (uid=${Authorization})
```

Within the mapping entry attribute `dsSearchFilter`, you can use placeholders of the format `${`*header*`}` where *header* is the name of an HTTP header.

The following are the most common headers used in DSML mappings.

`${Authorization}`	This string is replaced with the user name contained in an HTTP Authorization header. An authorization header contains both a username and its password, but only the user name is substituted in this placeholder.
`${From}`	This string is replaced with the email address that might be contained in an HTTP From header.
`${host}`	This string is replaced with the hostname and port number in the URL of the DSML request, which are those of the server.

To have DSML requests perform a different kind of identity mapping, define a new identity mapping for HTTP headers.

▼ To Define a New Identity Mapping for HTTP Headers

1. **Edit the default DSML-over-HTTP identity mapping or create custom mappings for this protocol.**

   The mapping entries must be located below the entry `cn=HTTP-BASIC,cn=identity mapping,cn=config`.

   Use the `ldapmodify` command to add this entry from the command line, as described in "Adding Entries Using ldapmodify" on page 79.

2. **Restart Directory Server for your new mappings to take effect.**

   Custom mappings are evaluated first. If no custom mapping is successful, the default mapping is evaluated. If all mappings fail to determine the bind DN for the DSML request, the DSML request is forbidden and rejected (error 403).

Setting the Server as Read-Only

Each suffix in your directory can be placed in read-only mode independently and can return a specific referral if one is defined. Directory Server also provides a read-only mode for the server that applies to all suffixes and can return a global referral when one is defined.

The server read-only mode is designed to allow administrators to prevent modifications to the directory contents while performing tasks such as reindexing the suffixes. For this reason, server read-only mode does not apply to the following configuration branches:

- `cn=config`
- `cn=monitor`
- `cn=schema`

These branches should be protected at all times by access control instructions (ACIs) against modifications by non-administration users, regardless of the read-only setting (see Chapter 6, "Directory Server Access Control"). Global read-only mode prevents update operations on all other suffixes in the directory, including update operations initiated by the Directory Manager.

Read-only mode also interrupts replication on a suffix if it is enabled. A master replica no longer has any changes to replicate, although it continues to replicate any changes that were made before read-only mode was enabled. A consumer replica does not receive updates until read-only mode is disabled. A master in a multi master replication environment does not have any changes to replicate and is not able to receive updates from the other masters.

▼ To Enable or Disable the Server Read-Only Mode

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Enable the global read-only mode.**

```
$ dsconf set-server-prop -h host -p port read-write-mode:read-only
```

2 **When you are ready, disable the read-only mode.**

```
$ dsconf set-server-prop -h host -p port read-write-mode:read-write
```

Configuring Memory

This section provides information about managing different types of memory. For a description of the different types of cache and for information about cache tuning, see Chapter 8, "Directory Server Data Caching," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

Priming Caches

To prime caches means to fill the caches with data so that subsequent Directory Server behavior reflects normal operational performance, rather than ramp-up performance. Priming caches is useful for arriving at reproducible results when benchmarking, and for measuring and analyzing potential optimizations.

If possible, do not actively prime the caches. Let the caches be primed by normal or typical client interaction with Directory Server before you measure performance.

Tools for priming database cache can be found at `http://www.slamd.com`.

▼ To Modify Database Cache

Caution – Modifying cache can severely impact server performance. Use caution when modifying cache.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Obtain the current database cache level.**

   ```
   $ dsconf get-server-prop -h host -p port db-cache-size
   ```

2. **Change the database cache level.**

   ```
   $ dsconf set-server-prop -h host -p port db-cache-size:size
   ```

   where *size* can be expressed in gigabytes (`G`), megabytes (`M`), kilobytes (`k`) or bytes (`b`). The size you specify must be supported by your machine.

▼ To Monitor Database Cache

The default level of cache at installation is suited to a test environment, not a production environment. For tuning purposes, you might want to monitor the database cache for your server.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Monitor database cache.**

  ```
  $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
   -b "cn=monitor,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
  ```

  If the database cache size is large enough and it has been primed, the hit ratio (`dbcachehitratio`) should be high. In addition, the number of pages that are read in (`dbcachepagein`) and the clean pages that are written out (`dbcacheroevict`) should be low. Here, "high" and "low" are relative to the deployment constraints.

▼ To Monitor Entry Cache

For tuning purposes, you might want to check the entry cache for one or more suffixes. Use this procedure to view the entry cache levels.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Monitor entry cache.**

  ```
  $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
   -b "cn=monitor,cn=db-name,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
  ```

  If the entry cache for a suffix is large enough to hold most of the entries in the suffix and if the cache is primed, the hit ratio (`entrycachehitratio`) should be high.

If you have primed the cache, you will see that as the previously empty entry cache fills, entry cache size (`currententrycachesize`) approaches the maximum entry cache size (`maxentrycachesize`). Ideally, the size in entries (`currententrycachecount`) should be either equal to or very close to the total number of entries in the suffix (`ldapentrycachecount`).

▼ To Modify Entry Cache

Caution – Modifying cache can severely impact server performance. Use caution when modifying cache.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Obtain the current entry cache level.**

```
$ dsconf get-suffix-prop -h host -p port suffix-DN entry-cache-count entry-cache-size
```

2 **Change the entry cache count.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-count:integer
```

where *integer* is the number of entries to be stored in the cache.

3 **Change the entry cache size.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-size:size
```

where *size* is the cache size expressed in gigabytes (`G`), megabytes (`M`), kilobytes (`k`) or bytes (`b`). The size you specify must be supported by your machine.

▼ To Configure Heap Memory Threshold

You can set threshold values for the dynamic memory footprint. You might need to set this threshold when Directory Server is running on a machine where resources are shared or sparse.

For information about memory sizing, see "Directory Server and Memory" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

Note – This threshold can only be set on Solaris and Linux platforms.

Note – By default, the `heap-high-threshold-size` and `heap-low-threshold-size` properties are `undefined`.

1. **Set the maximum heap high memory threshold.**

   ```
   $ dsconf set-server-prop -h host -p port heap-high-threshold-size:value
   ```

   where *value* is either `undefined` or a memory size expressed in gigabytes (`G`), megabytes (`M`), kilobytes (`k`) or bytes (`b`). The size you specify must be supported by your machine.

   For recommendations on the values to use for `heap-high-threshold-size` , see the `server(5dsconf)` man page.

2. **Optionally, set the maximum heap low memory threshold .**

   ```
   $ dsconf set-server-prop -h host -p port heap-low-threshold-size:value
   ```

   where *value* is either `undefined` or a memory size expressed in gigabytes (`G`), megabytes (`M`), kilobytes (`k`) or bytes (`b`). The size you specify must be supported by your machine.

   For recommendations on the values to use for `heap-low-threshold-size` , see the `server(5dsconf)` man page.

Setting Resource Limits For Each Client Account

You can control search operation resource limits on the server for each client account. You set such limits in operational attributes on an account, and Directory Server then enforces them based on the account a client uses to bind to the directory.

The following limits can be set:

- The look-through limit specifies the maximum number of entries examined for a search operation.
- The size limit specifies the maximum number of entries returned in response to a search operation.
- The time limit specifies the maximum time spent processing a search operation.
- The idle timeout specifies the maximum time a client connection can remain idle before the connection is dropped.

Note – The Directory Manager can use unlimited resources by default.

The resource limits that you set on specific user accounts take precedence over the resource limits set in the server-wide configuration. This section provides information about setting resource limits for each account.

The examples given in this section set resource limits directly in the attributes of the entry. You can also set resource limits on account using the Class of Service (CoS) mechanism. The CoS mechanism generates computed attributes as an entry is retrieved for a client application. For more information about defining CoS, see "Class of Service" on page 240.

▼ To Configure Search Limit

If you want to define the search limit that is used by the `nsslapd` process, refer to the following procedure:

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Use the `dsconf get-server-prop` command to read the resource limit server properties.

```
$ dsconf get-server-prop -h host -p port look-through-limit search-size-limit \
 search-time-limit idle-timeout
look-through-limit  :  5000
search-size-limit   :  2000
search-time-limit   :  3600
idle-timeout        :  none
```

The output shows that searches look through a maximum of 5000 entries, return a maximum of 2000 entries, and use a maximum of one hour (3600 seconds) of server time to process the search.

2 Change the look-through limit.

```
$ dsconf set-server-prop -h host -p port look-through-limit:integer
```

where *integer* is the maximum number of entries examined for a search operation.

3 Change the search size limit.

```
$ dsconf set-server-prop -h host -p port search-size-limit:integer
```

where *integer* is the maximum number of entries returned by a search operation.

4 Change the search time limit.

```
$ dsconf set-server-prop -h host -p port search-time-limit:integer
```

where *integer* is the maximum time spent processing a search operation.

5 Change the idle timeout.

```
$ dsconf set-server-prop -h host -p port idle-timeout:integer
```

where *integer* is the maximum time a client connection can remain idle before the connection is dropped.

◆ ◆ ◆ CHAPTER 4

Directory Server Entries

This chapter discusses how to manage the data entries in your directory. It also describes how to set referrals, encrypt attribute values, and compress entries.

When planning a directory deployment, you need to characterize the types of data that the directory will contain. Read the relevant chapters in the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide* before creating entries and modifying the default schema.

You cannot modify your directory unless the appropriate access control instructions (ACIs) have been defined. For further information, see Chapter 6, "Directory Server Access Control."

This chapter covers the following topics:

Managing Entries

The best way to manage entries depends on the context:

- If you mostly use DSCC for administration and you want to search or modify just a few entries, use DSCC. For more information about DSCC, see "Directory Service Control Center Interface" on page 537.
- If you want to search or modify a large number of entries, use the command-line utilities `ldapmodify` and `ldapdelete`.

Managing Entries Using DSCC

DSCC enables you to view all readable unencrypted attributes of an entry and to edit its writable attributes. It also enables you to add and remove attributes, set multi-valued attributes, and manage the object classes of the entry. For more information about how to use DSCC to manage entries, see the DSCC online help. For more information about DSCC in general, see "Directory Service Control Center Interface" on page 537.

Extending Entries Using DSCC

You can add or edit a directory entry directly through DSCC on the Entry Management tab page for a Directory Server instance. There are buttons that launch wizards for adding and editing entries.

The following procedure explains how to extend entries, adding additional user-defined attributes to existing entries. For example, a new application accessing the directory requires that you store additional information on each entry, and you need to create a few entries for testing purposes.

▼ To Extend Entries Using DSCC

1. **Use the schema wizard to set up a user-defined object class that specifies the attributes that you can add to the entry.**

   Click the link to the Directory instance; click the Schema tab; scroll to User-Defined Object Classes, and click the Add button to open the wizard.

   Alternatively, you can update the directory schema over LDAP or by editing configuration files. For more information, see Chapter 11, "Directory Server Schema."

   You must create the object class, because adding an object class attribute value to an entry is the LDAP way of extending the list of attributes the entry can have.

2. **From the Entry Management tab, and find the entry to update.**

3. **When you edit the entry, use the Text View.**

   The form-based editor shows you all the attributes you can edit, but not object classes you can add to extend the entry.

4. **In the Text View, add the object class and attributes you want in LDIF format.**

   For example, if you extend the schema with `example-objectclass` that allows the `example-attribute` attribute to add to the LDIF.

   ```
   dn: uid=bjensen,ou=People,dc=example,dc=com
   cn: Babs Jensen
   mail: bjensen@example.com
   ```

```
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: example-objectclass
sn: Jensen
uid: bjensen
example-attribute: Extended entry
```

The Text View editor has a check routine built in, so you can check that your edits are valid with a single click. When everything is the way you want it, apply your changes.

Managing Entries Using `ldapmodify` and `ldapdelete`

The `ldapmodify` and `ldapdelete` command-line utilities provide full functionality for adding, editing, and deleting your directory contents. You can use these utilities to manage both the configuration entries of the server and the data in the user entries. The utilities can also be used to write scripts to perform bulk management of one or more directories.

The `ldapmodify` and `ldapdelete` commands are used in procedures throughout this book. The following sections describe the basic operations that you will need to perform procedures. For more information about the `ldapmodify` and `ldapdelete` commands, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

Input to the command-line utilities is always in LDIF, and it can be provided either directly from the command-line or through an input file. The following section provides information about LDIF input, and subsequent sections describe the LDIF input for each type of modification.

For information about formatting LDIF input correctly, see the "Guidelines for Providing LDIF Input" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The following sections describe these basic operations:

Adding Entries Using ldapmodify

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Note – Ensure that you use the `ldapmodify` utility provided with the Directory Server Enterprise Edition software.

You can add one or more entries to the directory by using the `-a` option of `ldapmodify`. The following example creates a structural entry to contain users and then creates a user entry:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: ou=People,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Babs Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: secret
```

The `-D` and `-w` options give the bind DN and password, respectively, of a user with permissions to create these entries. The `-a` option indicates that all entries in the LDIF will be added. Then each entry is listed by its DN and its attribute values, with a blank line between each entry. The `ldapmodify` utility creates each entry after it is entered, and the utility reports any errors.

By convention, the LDIF of an entry lists the following attributes:

1. The DN of the entry.
2. The list of object classes.
3. The naming attribute (or attributes). This is the attribute used in the DN, and it is not necessarily one of the required attributes.
4. The list of required attributes for all object classes.
5. Any allowed attributes that you want to include.

When typing a value for the `userPassword` attribute, provide the clear text version of the password. The server will encrypt this value and store only the encrypted value. Be sure to limit read permissions to protect clear passwords that appear in LDIF files.

You can also use an alternate form of the LDIF that does not require the `-a` option on the command line. The advantage of this form is that you can combine entry addition statements and entry modification statements, as shown in the following example.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Barbara Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: secret
```

The `changetype: add` keyword indicates that the entry with the given DN should be created with all of the subsequent attributes. All other options and LDIF conventions are the same as explained earlier in this section.

In both examples, you can use the `-f` *filename* option to read the LDIF from a file instead of from the terminal input. The LDIF file must contain the same format as used for the terminal input, depending upon your use of the `-a` option.

Modifying Entries Using `ldapmodify`

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Note – Ensure that you use the `ldapmodify` utility that is provided as a part of the Directory Server Enterprise Edition software.

Use the `changetype: modify` keyword to add, replace, or remove attributes and their values in an existing entry. When you specify `changetype: modify`, you must also provide one or more change operations to indicate how the entry is to be modified. The three possible LDIF change operations are shown in the following example:

```
dn: entryDN
changetype: modify
add: attribute
attribute: value...
-
replace: attribute
attribute: newValue...
-
delete: attribute
[attribute: value]
...
```

Use a hyphen (`-`) on a line to separate operations on the same entry, and use a blank line to separate groups of operations on different entries. You can also give several *attribute*: *value* pairs for each operation.

Adding an Attribute Value

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

The following example shows how you can use the same `add` LDIF syntax to add values to existing multi-valued attribute and to attributes that do not yet exist:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: cn
cn: Babs Jensen
-
add: mobile
mobile: (408) 555-7844
```

This operation might fail and the server will return an error if any of the following are true:

- The given value already exists for an attribute.
- The value does not follow the syntax defined for the attribute.
- The attribute type is not required or allowed by the entry's object classes.
- The attribute type is not multi-valued and a value already exists for it.

Using the Binary Attribute Subtype

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

The *attribute*`;binary` subtype indicates that attribute values must be transported over LDAP as binary data, regardless of their actual syntax. This subtype is designed for complex syntax that does not have LDAP string representations, such as `userCertificate`. The binary subtype should not be used outside of this purpose.

When used with the `ldapmodify` command, appropriate subtypes can be added to attribute names in any of the LDIF statements.

To enter a binary value, you may type it directly in the LDIF text or read it from another file. The LDIF syntax for reading it from a file is shown in the following example:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
version: 1
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: userCertificate;binary
userCertificate;binary:< file:///local/cert-file
```

To use the `:<` syntax to specify a file name, you must begin the LDIF statement with the line `version: 1`. When `ldapmodify` processes this statement, it will set the attribute to the value that is read from the entire contents of the given file.

By default, the search returns the binary attributes when used with the `;binary` option. Set the `compat-flag` to `norfc4522` to disable `rfc4522` compliance.

Adding an Attribute With a Language Subtype

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Language and pronunciation subtypes of attributes designate localized values. When you specify a language subtype for an attribute, the subtype is added to the attribute name as follows:

attribute`;lang-`*CC*

where *attribute* is an existing attribute type, and *cc* is the two-letter country code to designate the language. You may optionally add a pronunciation subtype to a language subtype to designate a phonetic equivalent for the localized value. In this case the attribute name is as follows:

attribute`;lang-`*CC*`;phonetic`

To perform an operation on an attribute with a subtype, you must explicitly match its subtype. For example, if you want to modify an attribute value that has the `lang-fr` language subtype, you must include `lang-fr` in the modify operation as follows:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34, rue de la Paix
```

Note – If the attribute value contains non-ASCII characters, they must be UTF-8 encoded.

Modifying Attribute Values

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

The following example shows how to change the value of an attribute by using the `replace` syntax in LDIF:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: sn
sn: Morris
-
replace: cn
cn: Barbara Morris
cn: Babs Morris
```

All current values of the specified attributes are removed, and all given values are added.

After changing an attribute value, you can use the `ldapsearch` command to verify the change.

Trailing Spaces in Attribute Values

When you modify an attribute value, do not unintentionally include trailing spaces at the end of the value. The trailing spaces are stored in the server as part of the attribute's value, and leads to unexpected values being stored

When you verify the change using DSCC or the `ldapsearch` command, the value you see might be plain text or some other unexpected value. This depends on which Directory Server client you use.

Deleting an Attribute Value

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

The following example shows how to delete an attribute entirely and to delete only one value of a multi valued attribute:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: facsimileTelephoneNumber
-
delete: cn
cn: Babs Morris
```

When using the `delete` syntax without specifying an *attribute: value* pair, all values of the attribute are removed. If you specify an *attribute: value* pair, only that value is removed.

Modifying One Value of a Multi Valued Attribute

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

To modify one value of a multi valued attribute with the `ldapmodify` command, you must perform two operations as shown in the following example:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: mobile
mobile: (408) 555-7845
-
add: mobile
mobile: (408) 555-5487
```

Deleting Entries Using `ldapdelete`

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Note – Ensure that you use the `ldapdelete` utility that is provided as a part of the Directory Server Enterprise Edition software.

Use the `ldapdelete` command-line utility to delete entries from the directory. This utility binds to the directory server and deletes one or more entries based on their DN. You must provide a bind DN that has permission to delete the specified entries.

You cannot delete an entry that has children. The LDAP protocol forbids the situation where child entries would no longer have a parent. For example, you cannot delete an organizational unit entry unless you have first deleted all entries that belong to the organizational unit.

The following example shows only one entry in the organizational unit. This entry and then its parent entry can be deleted.

```
$ ldapdelete -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
uid=bjensen,ou=People,dc=example,dc=com
ou=People,dc=example,dc=com
```

Deleting Entries Using `ldapmodify`

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Note – Ensure that you use the `ldapmodify` utility that is provided as a part of the Directory Server Enterprise Edition software.

When using the `ldapmodify` utility, you can also use the `changetype: delete` keywords to delete entries. All of the same limitations apply as when using `ldapdelete`, as described in the previous section. The advantage of using LDIF syntax for deleting entries is that you can perform a mix of operations in a single LDIF file.

The following example performs the same delete operations as the previous example:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: delete

dn: ou=People,dc=example,dc=com
changetype: delete
```

Searching Entries Using `ldapsearch`

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Note – Ensure that you use the `ldapsearch` utility that is provided as a part of the Directory Server Enterprise Edition software.

You can use the `ldapsearch` command-line utility to locate and retrieve directory entries.

For more information about using `ldapsearch`, common `ldapsearch` options, accepted formats, and examples, refer to *Sun Directory Server Enterprise Edition 7.0 Reference*.

▼ To Move or Rename an Entry Using `ldapmodify`

This procedure uses the modify DN operation. Before starting this operation, ensure that you are familiar with the section "Guidelines and Limitations for Using the Modify DN Operation" on page 89.

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

Note – When modifying the DNs of entries that are a `uniquemember` of a group, you must have the referential integrity plug-in enabled. Referential integrity ensures that the group members get adjusted when the entry is moved. For information about how to enable and configure the referential integrity plug-in, see "To Configure the Referential Integrity Plug-In" on page 252.

1 **If you are moving an entry from one parent to another, extend ACI rights on the parent entries.**

- On the current parent entry of the entry to be moved, ensure that the ACI allows the `export` operations by using the syntax `allow (export ...)`
- On the future parent entry of the entry to be moved, ensure that the ACI allows the `import` operations. by using the syntax `allow (import ...)`

For information about using ACIs, see Chapter 6, "Directory Server Access Control."

2 **Ensure that the modify DN operation is enabled globally, or at least for the suffix or suffixes that will be affected by the move operation.**

To ensure compatibility with previous releases of Directory Server, the modify DN operation is not enabled by default.

If you have already enabled the modify DN operation previously, go to the next step.

To enable the modify DN operation globally for a server, use this command:

```
$ dsconf set-server-prop -h host -p port moddn-enabled:on
```

3 Run the `ldapmodify` command.

This step uses the modify DN operation. Do one of the following:

- Move the entry.

  For example, the following command moves the entry `uid=bjensen` from the subtree for contractors, `ou=Contractors,dc=example,dc=com` to the subtree for employees, `ou=People,dc=example,dc=com`:

  ```
  $ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
  Enter bind password:
  dn: uid=bjensen,ou=Contractors,dc=example,dc=com
  changetype: modrdn
  newrdn: uid=bjensen
  deleteoldrdn: 0
  newsuperior: ou=People,dc=example,dc=com
  ```

- Rename the entry.

  For example, the following command renames the entry `uid=bbjensen` to `uid=bjensen`:

  ```
  $ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
  Enter bind password:
  dn: uid=bbjensen,ou=People,dc=example,dc=com
  changetype: modrdn
  newrdn: uid=bjensen
  deleteoldrdn: 1
  ```

Pay attention to the following attributes when writing the LDIF statement:

- `dn` - Specifies the entry to rename or move.
- `changetype: modrdn` - Specifies that a modify DN operation is to be used.
- `newrdn` - Gives the new naming attribute.
- `deleteoldrdn` - Indicates whether the previous naming attribute should be removed from the entry (`1` is yes, `0` is no).

  Note that you cannot remove a naming attribute from the entry if that attribute is obligatory in the entry definition.
- `newsuperior` - Specifies the new superior attribute of the entry.

For information about the `ldapmodify` command and its options, see the `ldapmodify(1)` man page.

4 If you encounter resource limit errors when moving or renaming subtrees that contain a large number of entries, increase the number of locks that can be used by the database.

`$ dsconf set-server-prop -h` *host* `-p` *port* `db-lock-count:`*value*

If you modify this property, you must restart the server for the change to take effect.

Guidelines and Limitations for Using the Modify DN Operation

When you use the modify DN operation, as described in the previous section, use the guidelines described in the following sections.

General Guidelines for Using the Modify DN Operation

- Do not use the modify DN operation to move an entry from one suffix to another suffix, or to rename or move the root suffix.
- Do not use the `entryid` operational attribute in your application because it is reserved for internal use only. The `entryid` attribute of an entry can change when an entry is moved.
- Enable the modify DN operation globally for all suffixes on a server, or individually on each suffix where you wish to run the operation. By default the modify DN operation is disabled.
- Extend the ACI rights on each suffix where you wish to run the modify DN operation. The `Import` access right allows an entry to be imported to the specified DN. The `Export` access right allows an entry to be exported from the specified DN.
- Before performing a modify DN operation, ensure that the operation would not break client authentication. If you move an entry that refers to a client certificate, client authentication will break. After moving an entry, validate your certificates.
- Before performing a modify DN operation, ensure that the operation would not break your application. The rename or move of an entry can affect several suffixes, or can change the following characteristics of the entry:
  - The scope of a filtered role of an entry.
  - The nested role of an entry, where the nested role contains a filtered role.
  - The dynamic group membership of an entry.

Guidelines for Using the Modify DN Operation With Replication

Caution – Using the modify DN operation without complying with the following requirements can break replication and bring down your directory service.

- Enable the modify DN operation on all servers in your replication topology. If the modify DN operation is supported on the master server but not on the consumer server, replication will fail. A message similar to the following will be written to the error log on the supplier server:

  ```
  Unable to start a replication session with MODDN enabled
  ```

To restart replication, reconfigure the replication topology to enable the modify DN operation on all servers. and then start a replication session in one of the following ways:

- By following the instructions in "To Force Replication Updates" on page 287.
- By changing an entry on the supplier server. The change is replicated to the consumer servers.

- Enable and configure the referential integrity plug-in on all master replicas in the topology. This action ensures that the server maintains referential integrity for groups and roles. For information about how to enable and configure the referential integrity plug-in, see "To Configure the Referential Integrity Plug-In" on page 252.

  After performing a modify DN operation, allow time for the referential integrity plug-in to replicate its changes.

Grouping Entries for Simplified Management

You can simplify entry management by associating related entries in groups. The group mechanism makes it easy to retrieve a list of entries that are members of a given group and set access permissions for a whole group.

Entries can be managed as members of dynamic and static groups. Static groups are suitable for groups with few members, such as a group of directory administrators. A dynamic group specifies one or more URL search filters, so the dynamic group membership is defined each time these search filters are evaluated.

You can retrieve a list of all the static groups a given user is a member of by using the dynamic `isMemberOf` attribute. This attribute is located in the user entry and in nested group entries and holds the DNs of the static groups to which the member belongs. For example, `Kirsten Vaughan` is a new system administrator in the human resources department. Her entry shows that she is a member of both the System Administrators group and the HR Managers group.

```
$ ldapsearch -b "dc=example,dc=com" uid=kvaughan isMemberOf

uid=kvaughan, ou=People, dc=example,dc=com
isMemberOf: cn=System Administrators, ou=Groups, dc=example,dc=com
isMemberOf: cn=HR Managers,ou=groups,dc=example,dc=com
```

Membership testing for group entries has been improved. These improvements remove some of the previous restrictions on static groups, specifically the restriction on group size. This performance improvement is only effective after the group entry has been loaded into the entry cache.

Compressing Entries

To reduce the database entry size, entities can be compressed. The compressed entry results in improved performance through reducing the number of overflow pages in large databases.

▼ To Compress the Size of Entries in Database

1 **Configure `compressed-entries` to specify the entries that need to be compressed.**

```
$ dsconf set-suffix-prop -p port-number dc=example,dc=com compressed-entries:all
```

For more information, see `compressed-entries`(5dsconf)

2 **Configure `compression-mode` to specify the compression technique.**

```
$ dsconf set-suffix-prop -p port-number dc=example,dc=com compression-mode:DSZ
```

For more information, see `compression-mode`(5dsconf)

Setting Referrals

You can use referrals to tell client applications which server to contact if the information is not available locally. Referrals are pointers to a remote suffix or entry that Directory Server returns to the client, in place of a result. The client must then perform the operation again on the remote server named in the referral.

Redirection occurs in three cases:

- When a client application requests an entry that does not exist on the local server, and the server has been configured to return the default referral.
- When an entire suffix has been disabled for maintenance or security reasons.

  The server will return the referrals defined by that suffix. The suffix-level referrals are described in "Setting Referrals and Making a Suffix Read-Only" on page 50. Read-only replicas of a suffix also return referrals to the master servers when a client requests a write operation.
- When a client specifically accesses a smart referral.

  A *smart referral* is an entry that you create. The server will return the referral that the smart referral defines.

In all cases, a referral is an LDAP URL that contains the host name, port number, and optionally a DN on another server. For example, `ldap://east.example.com:389`.

For conceptual information about how you can use referrals in your directory deployment, see the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

The following sections describe the procedures for setting your directory's default referrals and for creating and defining smart referrals.

Setting the Default Referrals

Default referrals are returned to client applications that submit operations on a DN that is not contained on a suffix maintained by your Directory Server. The server will return all referrals that are defined, but the order in which they are returned is not defined.

▼ To Set a Default Referral

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Use the `dsconf` command-line utility to set one or more default referrals.**

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN referral-url:referral-URL
  ```

  For example:

  ```
  $ dsconf set-server-prop -h host1 -p 1389 dc=example,dc=com \
   referral-url:ldap://east.example.com:1389
  ```

Setting Smart Referrals

Smart referrals allow you to map a directory entry or a directory tree to a specific LDAP URL. Using smart referrals, you can refer client applications to a specific server or to a specific entry on a specific server.

Often, a smart referral points to an actual entry with the same DN on another server. However, you may define the smart referral to any entry on the same server or on a different server. For example, you can define the entry with the following DN to be a smart referral:

```
uid=bjensen,ou=People,dc=example,dc=com
```

The smart referral points to another entry on the server `east.example.com`:

```
cn=Babs Jensen,ou=Sales,o=east,dc=example,dc=com
```

The way the directory uses smart referrals conforms to the standard specified in section 4.1.10 of RFC 4511 (`http://www.ietf.org/rfc/rfc4511.txt`).

▼ To Create and Modify a Smart Referral

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 To create a smart referral, create an entry with `referral` and `extensibleObject` object classes.

The `referral` object class allows the `ref` attribute that is expected to contain an LDAP URL. The `extensibleObject` object class allows you to use any schema attribute as the naming attribute, in order to match the target entry.

For example, to define the following entry to return a smart referral instead of the entry `uid=bjensen`, use this command:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: referral
uid: bjensen
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Sales,o=east,dc=example,dc=com
```

Note – Any information after a space in an LDAP URL is ignored by the server. Thus, you must use `%20` instead of spaces in any LDAP URL that you intend to use as a referral. Other special characters must be escaped.

After you have defined the smart referral, modifications to the `uid=bjensen` entry will actually be performed on the `cn=Babs Jensen` entry on the other server. The `ldapmodify` command will automatically follow the referral, for example:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: telephoneNumber
telephoneNumber: (408) 555-1234
```

2 (Optional) To modify the smart referral entry, use the `-M` option of `ldapmodify`:

```
$ ldapmodify -M -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: ref
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Marketing,o=east,dc=example,dc=com
```

Checking Valid Attribute Syntax

Directory Server allows you to check the integrity of your attributes whenever you perform the following operations:

- Importing data using `dsadm import` or `dsconf import`.
- Using LDAP or DSML to add entries, modify entries, or modify the DN of an entry.

The checks ensure that the attribute values conform to IETF recommendations. All nonconforming attributes are rejected and logged in the errors log. The log messages include the connection and operation ID, if applicable.

By default, the server does not automatically check the syntax of the previously mentioned operations. If you want to turn syntax checking on, use the following procedure.

Note – Syntax checking is not the same as schema checking. For information about schema checking, see "Managing Schema Checking" on page 299.

▼ To Turn On Automatic Syntax Checking

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **To turn on automatic syntax checking, use this command:**

```
$ dsconf set-server-prop -h host -p port check-syntax-enabled:on
```

Tracking Modifications to Directory Entries

By default, the server maintains special attributes for newly created or modified entries, as specified in the LDAP v3 specification. These special attributes are stored on the entry in the suffix and include the following:

- `creatorsName` — the DN of the user who initially created the entry.
- `createTimestamp` — the timestamp for when the entry was created, in GMT format.
- `modifiersName` — the DN of the user who last modified the entry.
- `modifyTimestamp` — the timestamp for when the entry was modified, in GMT format.

▼ To Turn Off Entry Modification Tracking

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Caution – Turning off entry modification tracking results in non-compliant data. As many applications rely on these attributes and as disabling this feature results in only minimal performance gains, we recommend that you do not turn off entry modification tracking.

- **Turn off entry modification tracking for the server.**

```
$ dsconf set-server-prop -h host -p port suffix-DN mod-tracking-enabled:off
```

Encrypting Attribute Values

Attribute encryption protects sensitive data while it is stored in the directory. Attribute encryption allows you to specify that certain attributes of an entry are stored in an encrypted format. This prevents data from being readable while stored in database files, backup files, and exported LDIF files.

The only data that is saved on the disk is encrypted. Not all the data that is displayed under the Entry Management panel of the console or as an output of the `ldapsearch` command, is encrypted.

With this feature, attribute values are encrypted before they are stored in the Directory Server database, and decrypted back to their original value before being returned to the client. You must use access controls to prevent clients from accessing such attributes without permission, and SSL to encrypt all exchanges between the client and Directory Server. For an architectural overview of data security in general and attribute encryption in particular, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

Attribute encryption is active only when SSL is configured and enabled on the server. However, no attributes are encrypted by default. Attribute encryption is configured at the suffix level, which means that an attribute is encrypted in every entry in which it appears in the suffix. If you want to encrypt an attribute in an entire directory, you must enable encryption for that attribute in every suffix.

Caution – Attribute encryption affects all data and index files associated with a suffix. The only attributes are encrypted that are changed after the attribute encryption is activated. Existing attributes will remain unchanged.

To apply encryption to all the data, you *must* first export its contents, make the configuration change, and then re-import the contents. DSCC can help you perform these steps. For more information about using DSCC, see "Directory Service Control Center Interface" on page 537.

For additional security, when turning on encryption for any attribute, you should manually delete the database cache files and database log file that might still contain unencrypted values. The procedure for deleting these files is described in "To Configure Attribute Encryption" on page 98.

You should enable any encrypted attributes before loading or creating data in a new suffix.

If you choose to encrypt an attribute that some entries use as a naming attribute, values that appear in the DN will not be encrypted. Values that are stored in the entry will be encrypted.

Even though you can select the `userPassword` attribute for encryption, no real security benefit is realized unless the password needs to be stored in the clear. Such is the case for DIGEST-MD5 SASL authentication. If the password already has an encryption mechanism defined in the password policy, further encryption provides little additional security, but, it will impact the performance of every bind operation.

When in storage, encrypted attributes are prefaced with a cipher tag that indicates the encryption algorithm used. An encrypted attribute using the DES encryption algorithm would appear as follows:

```
{CKM_DES_CBC}3hakc&jla+=snda%
```

When importing data online with a view to encrypting it, you will already have provided the key database password to authenticate to the server and will not be prompted a second time. If you are importing data offline, Directory Server will prompt you for the password before it allows you to encrypt the data you are importing. When decrypting data (a more security-sensitive operation), Directory Server automatically prompts you for the key database password, regardless of whether the export operation is online or offline. This provides an additional security layer.

Note – As long as the certificate or private key does not change, the server will continue to generate the same key. Thus, data can be transported (exported then imported) from one server instance to another, provided both server instances have used the same certificate.

Attribute Encryption and Performance

While attribute encryption offers increased data security, it does impact system performance. Think carefully about which attributes require encryption, and encrypt only those attributes that you consider to be particularly sensitive.

Because sensitive data can be accessed directly through index files, the index keys that correspond to the encrypted attributes must be encrypted to ensure that the attributes are fully protected. Given that indexing already has an impact on Directory Server performance (without the added cost of encrypting index keys), configure attribute encryption *before* data is imported or added to the database for the first time. This procedure will ensure that encrypted attributes are indexed as such from the outset.

Attribute Encryption Usage Considerations

Consider the following when implementing the attribute encryption feature:

- As a general best practice when modifying attribute encryption configuration, you should export your data, make the configuration changes, and then import the newly configured data.

  This will ensure that all configuration changes are taken into account in their entirety, without any loss in functionality. Failing to do so could result in some functionality loss and thus compromise the security of your data.

- Modifying attribute encryption configuration on an existing database can have a significant impact on system performance.

  For example, imagine that you have a database instance with existing data. The database contains previously stored entries with an attribute called `mySensitiveAttribute`. The value of this attribute is stored in the database and in the index files in clear text, . If you later decide to encrypt the `mySensitiveAttribute` attribute, all the data in the database instance must be exported and re-imported into the database to ensure that the server updates the database and index files with the attribute encryption configuration. The resulting performance impact could have been avoided had the attribute been encrypted from the beginning.

- When exporting data in decrypted format, the export is refused if an incorrect password is used.

  To be able to use the `dsconf` command with the `-y` or `-–decrypt-attr` option, set the `set password prompt` mode `on` and choose a certificate database password as described in "Configuring the Certificate Database Password" on page 110.

  As a security measure, the server prompts users for passwords if they want to export data in decrypted format. Should users provide an incorrect password, the server refuses the decrypted export operation. Passwords can be entered directly or by providing the path to a file that contains the password. Note that this file has the same syntax as the SSL password file. See "Configuring the Certificate Database Password" on page 110.

- Algorithm changes are supported, but the result can be lost indexing functionality if they are not made correctly.

  To change the algorithm used to encrypt data, export the data, modify the attribute encryption configuration, and then import the data. If you do not follow this procedure, the indexes that were created on the basis of the initial encryption algorithm will no longer function.

  Because the encrypted attributes are prefaced with a cipher tag that indicates the encryption algorithm used, the internal server operations take care of importing the data. Directory Server therefore enables you to export data in encrypted form before making the algorithm change.
- Changing the server's SSL certificate results in your not being able to decrypt encrypted data.

  The server's SSL certificate is used by the attribute encryption feature to generate its own key, which is then used to perform the encryption and decryption operations. Thus, the SSL certificate is required to decrypt encrypted data. If you change the certificate without decrypting the data beforehand, you cannot decrypt the data. To avoid this, export your data in decrypted format, change the certificate, and then re-import the data.
- To transport data in encrypted format, that is, to export and import it from one server instance to another, both server instances must use the same certificate.

  For information, see "Encrypting Attribute Values" in the *Sun Directory Server Enterprise Edition 7.0 Administration Guide*.

▼ To Configure Attribute Encryption

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 If the suffix on which you want to configure attribute encryption contains any entries whatsoever, you must first export the contents of that suffix to an LDIF file.

If the suffix contains encrypted attributes and you plan to re-initialize the suffix using the exported LDIF file, you can leave the attributes encrypted in the exported LDIF .

2 To enable encryption for an attribute, use this command:

```
$ dsconf create-encrypted-attr -h host -p port suffix-DN attr-name cipher-name
```

where *cipher-name* is one of the following:

- `des` - DES block cipher
- `des3` - Triple-DES block cipher
- `rc2` - RC2 block cipher
- `rc4` - RC4 stream cipher

For example:

```
$ dsconf create-encrypted-attr -h host1 -p 1389 dc=example,dc=com uid rc4
```

3 **Initialize the suffix with an LDIF file as described in "Initializing a Suffix" on page 52.**

Note – If you are importing the LDIF file using the `dsadm import` command, you must use the `-y` option. The `dsconf import` command does not require to use the `-y` option.

As the file is loaded and the corresponding indexes are created, all values of the specified attributes will be encrypted.

◆ ◆ ◆ CHAPTER 5

Directory Server Security

Directory Server supports several mechanisms that provide secure and trusted communications over the network. LDAPS is the standard LDAP protocol that runs on top of the Secure Sockets Layer (SSL). LDAPS encrypts data and optionally uses certificates for authentication. When the term SSL is used in this chapter, it means the supported protocols SSL2, SSL3 and TLS 1.0.

Directory Server also supports the Start Transport Layer Security (Start TLS) extended operation to enable TLS on an LDAP connection that was originally not encrypted.

In addition, Directory Server supports the Generic Security Service API (GSSAPI) over the Simple Authentication and Security Layer (SASL). The GSSAPI allows you to use the Kerberos Version 5 security protocol on the Solaris Operating System (Solaris OS). An identity mapping mechanism then associates the Kerberos principal with an identity in the directory.

For additional security information, see the NSS web site at `http://www.mozilla.org/projects/security/pki/nss/`.

This chapter provides procedures for configuring security through SSL. For information about ACIs, see Chapter 6, "Directory Server Access Control." For information about user access and passwords, see Chapter 7, "Directory Server Password Policy."

This chapter covers the following topics:

Using SSL With Directory Server

The Secure Sockets Layer (SSL) provides encrypted communication and optional authentication between a Directory Server and its clients. SSL can be used over LDAP or with DSML-over-HTTP. SSL is enabled by default over LDAP, but if you are using DSML-over-HTTP, you can easily enable SSL. In addition, replication can be configured to use SSL for secure communications between servers.

Using SSL with simple authentication (bind DN and password) encrypts all data sent to and from the server. Encryption guarantees confidentiality and data integrity. Optionally, clients can use a certificate to authenticate to Directory Server or to a third-party security mechanism through the Simple Authentication and Security Layer (SASL). Certificate-based authentication uses public-key cryptography to prevent forgery and impersonation of either the client or the server.

Directory Server is capable of simultaneous SSL and non-SSL communications on separate ports. For security reasons, you can also restrict all communications to the LDAP secure port. Client authentication is also configurable. You can set client authentication to required or to allowed. This setting determines the level of security you enforce.

SSL enables support for the Start TLS extended operation that provides security on a regular LDAP connection. Clients can bind to the standard LDAP port and then use the Transport Layer Security protocol to secure the connection. The Start TLS operation allows more flexibility for clients, and can help simplify port allocation.

The encryption mechanisms provided by SSL are also used for attribute encryption. Enabling SSL allows you to configure attribute encryption on your suffixes, which protects data while it is stored in the directory. For more information, see "Encrypting Attribute Values" on page 95.

For additional security, you can set access control to directory contents through access control instructions (ACIs). ACIs require a specific authentication method and ensure that data can only be transmitted over a secure channel. Set the ACIs to complement your use of SSL and certificates. For more information, see Chapter 6, "Directory Server Access Control."

SSL is enabled by default over LDAP, and you can easily enable SSL for DSML-over-HTTP. In addition, there are some aspects of the SSL configuration that you might want to modify, as described in the following sections.

Managing Certificates

This section describes how to manage SSL certificates in Directory Server.

To run SSL on Directory Server, you must either use a self-signed certificate or a Public Key Infrastructure (PKI) solution.

The PKI solution involves an external Certificate Authority (CA). For a PKI solution, you need a CA-signed server certificate, which contains both a public key and a private key. This certificate is specific to one Directory Server. You also need a trusted CA certificate, which contains a public key. The trusted CA certificate ensures that all server certificates from your CA are trusted. This certificate is sometimes called a CA root key or root certificate.

Note – If you are using certificates for test purposes, you probably want to use self-signed certificates. However, in production, using self-signed certificates is not very secure. In production, use trusted Certificate Authority (CA) certificates.

The procedures in this section use the `dsadm` and `dsconf` commands. For information about these commands, see the `dsadm(1M)` and `dsconf(1M)` man pages.

This section provides the following information about configuring certificates on Directory Server:

▼ To View the Default Self-Signed Certificate

When a Directory Server instance is first created, it contains a default self-signed certificate. A *self-signed certificate* is a public and private key pair, where the public key is signed by the private key. A self-signed certificate is valid for 24 months.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **To view the default self-signed certificate, use this command:**

```
$ dsadm show-cert instance-path defaultCert
```

To Manage Self-Signed Certificates

When you create a Directory Server instance, a default self-signed certificate is automatically provided.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **To create a self-signed certificate with non-default settings, use this command:**

   ```
   $ dsadm add-selfsign-cert instance-path cert-alias
   ```

   Where *cert-alias* is a name that you provide to identify your certificate.

   To see all the options for this command, see the `dsadm`(1M) man page or the command-line help:.

   ```
   $ dsadm add-selfsign-cert --help
   ```

2. **When your self-signed certificate expires, stop the server instance and renew the certificate.**

   ```
   $ dsadm stop instance-path
   $ dsadm renew-selfsign-cert instance-path cert-alias
   ```

3. **Restart the server instance.**

   ```
   $ dsadm start instance-path
   ```

To Request a CA-Signed Server Certificate

This procedure explains how to request and install a CA-signed server certificate for use with Directory Server.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Generate a CA-signed server certificate request.**

   ```
   $ dsadm request-cert [-i] [-W cert-pwd-file] {-S DN | --name name [--org org] \
     [--org-unit org-unit] [--city city] [--state state] [--country country]} \
     [--phone PHONE] [--email EMAIL] [--dns DOMAIN] [--keysize KEYSIZE] \
     [--sigalg SIGALG] [-F format] [-o output-file] instance-path
   ```

   For example, to request a CA-signed server certificate for the Example company, use this command:

   ```
   $ dsadm request-cert --name host1 --org Example --org-unit Marketing \
    -o my_cert_request_file /local/dsInst
   ```

In order to completely identify the server, Certificate Authorities might require all of the attributes that are shown in this example. For a description of each attribute, see the `dsadm(1M)` man page.

When you request a certificate by using `dsadm request-cert`, the resulting certificate request is a binary certificate request unless you specify ASCII as output format. If you specify ASCII, the resulting certificate request is a PKCS #10 certificate request in PEM format. PEM is the Privacy Enhanced Mail format specified by RFCs 1421 through 1424 (`http://www.ietf.org/rfc/rfc1421.txt`) and is used to represent a base64-encoded certificate request in US-ASCII characters. The content of the request looks similar to the following example:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UBhMCVXMxEzARBgNVBAgTCkNBElGT1JOSUExLD
AqBgVBAoTI25ldHNjYXBlIGNvb11bmljYXRpb25zIGNvcnBvcmF0aWuMRwwGgYDV
QQDExNtZWxsb24umV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAUAA4GNADCBiQK
BgCwAbskGh6SKYOgHy+UCSLnm3ok3X3u83Us7u0EfgSLR0f+K41eNqqWRftGR83e
mqPLDOf0ZLTLjVGJaHJn4l1gG+JDf/n/zMyahxtV7+T8GOFFigFfuxJaxMjr2j7I
vELlxQ4IfZgwqCm4qQecv3G+N9YdbjveMVXW0v4XwIDAQABAAwDQYJKoZIhvcNAQ
EEBQADgYEAZyZAm8UmP9PQYwNy4Pmypk79t2nvzKbwKVb97G+MT/gw1pLRsuBoKi
nMfLgKp1Q38K5Py2VGW1E47/rhm3yVQrIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

You must save the request at a secure place for future reference. You may need the request for renewal.

2 Transmit the certificate request to your Certificate Authority, according to its procedures.

The process for obtaining your Certificate Authority certificate depends on the certificate authority that you use. Some commercial CAs provide a website that allows you to automatically download the certificate. Other CAs will send it to you in email upon request.

After you have sent your request, you must wait for the CA to respond with your certificate. Response time for your request varies. For example, if your CA is internal to your company, the CA might only take a day or two to respond to your request. If your selected CA is external to your company, the CA could take several weeks to respond to your request.

3 Save the certificate that you receive from the Certificate Authority.

Back up your certificates in a safe location. If you ever lose the certificates, you can reinstall them by using your backup file. You can save them in text files. The PKCS #11 certificate in PEM format looks similar to the following example:

```
-----BEGIN CERTIFICATE-----
MIICjCCAZugAwIBAgICCEEwDQYJKoZIhKqvcNAQFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoGlBhbG9a2FWaWxsZGwSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTW3FrZXJzICdSJyBVczEpMCcGAx1UEAxgVGVzdCBUXN0IFRlc3QgVGVz
dCBUZXN0IFlc3QgQ0EswHhcNOTgwMzEyMDIzMzUWhcNOTgwMzI2MDIzMpzU3WjBP
MQswCYDDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZN0b3J5VIFB1Ymxp
```

```
Y2F0aW9uczEWMB4QGA1UEAxMNZHVgh49dq2tLNvbjTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYkCQCksMR/aLGdfp4m0OiGgijG5KgOsyRNvwGYW7kfW+8mmijDtZaRjYNj
jcgpF3VnlbxbclX9LVjjNLC5737XZdAgEDozYwpNDARBglghkgBhvhCEAQEEBAMC
APAwHkwYDVR0jBBgwFAU67URjwCaGqZHUpSpdLxlzwJKiMwDQYJKoZIhQvcNAQEF
BQADgYEAJ+BfVem3vBOPBveNdLGfjlb9hucgmaMcQa9FA/db8qimKT/ue9UGOJqL
bwbMKBBopsDn56p2yV3PLIsBgrcuSoBCuFFnxBnqSiTS7YiYgCWqWaUA0ExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

To Add the CA-Signed Server Certificate and the Trusted CA Certificate

This procedure explains how to install the CA-signed server certificate and trusted CA certificates for use with Directory Server.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Add the CA-signed server certificate.

```
$ dsadm add-cert instance-path cert-alias cert-file
```

Where *cert-alias* is a name that you provide to identify your certificate, and *cert-file* is the text file that contains the PKCS #11 certificate in PEM format.

For example, to install a CA-signed server certificate, you might use a command similar to this:

```
$ dsadm add-cert /local/dsInst server-cert /local/safeplace/serv-cert-file
```

The certificate is now installed, but is not yet trusted. To trust the CA-signed server certificate, you must install the Certificate Authority certificate.

2 Add the trusted Certificate Authority certificate.

```
$ dsadm add-cert --ca instance-path cert-alias cert-file
```

The `--ca` option indicates that the certificate is a trusted Certificate Authority certificate.

For example, to install a trusted certificate from a Certificate Authority, you might use this command:

```
$ dsadm add-cert --ca /local/dsInst CA-cert /local/safeplace/ca-cert-file
```

3 (Optional) Verify your installed certificates.

- To list all server certificates and to display their validity dates and aliases, type:

  ```
  $ dsadm list-certs instance-path
  ```

  For example:

```
$ dsadm list-certs /local/ds1
Enter the certificate database password:
Alias       Valid from Expires on Self-   Issued by          Issued to
                                signed?
----------- ---------- ---------- ------- -----------------  -----------------
serverCert  2000/11/10 2011/02/10 n       CN=CA-Signed Cert, CN=Test Cert,
            18:13      18:13              OU=CA,O=com        dc=example,dc=com
defaultCert 2006/05/18 2006/08/18 y       CN=host1,CN=DS,    Same as issuer
            16:28      16:28              dc=example,dc=com
2 certificates found
```

By default, an instance of Directory Server contains a default server certificate called `defaultCert`. The text `Same as issuer` indicates that the default certificate is a self-signed server certificate.

- To list trusted CA certificates, type:

  ```
  $ dsadm list-certs -C instance-path
  ```

  For example:

  ```
  $ dsadm list-certs -C /local/ds1
  Enter the certificate database password:
  Alias   Valid from Expires on Self-   Issued by           Issued to
                                signed?
  ------- ---------- ---------- ------- -----------------   --------------
  CA-cert 2000/11/10 2011/02/10 y       CN=Trusted CA Cert, Same as issuer
          18:12      18:12              OU=CA,O=com
  1 certificate found
  ```

- To view the details of a certificate, including the certificate expiration date, type:

  ```
  $ dsadm show-cert instance-path cert-alias
  ```

  For example, to view a server certificate, type:

  ```
  $ dsadm show-cert /local/ds1 "Server-Cert"

  Enter the certificate database password:
  Certificate:
      Data:
          Version: 3 (0x2)
          Serial Number: 2 (0x2)
          Signature Algorithm: PKCS #1 MD5 With RSA Encryption
          Issuer:
              "CN=Server-Cert,O=Sun,C=US"
          Validity:
              Not Before: Fri Nov 10 18:12:20 2000
              Not After : Thu Feb 10 18:12:20 2011
          Subject:
              "CN=CA Server Cert,OU=ICNC,O=Sun,C=FR"
  ```

```
    Subject Public Key Info:
        Public Key Algorithm: PKCS #1 RSA Encryption
        RSA Public Key:
            Modulus:
                bd:76:fc:29:ca:06:45:df:cd:1b:f1:ce:bb:cc:3a:f7:
                77:63:5a:82:69:56:5f:3d:3a:1c:02:98:72:44:36:e4:
                68:8c:22:2b:f0:a2:cb:15:7a:c4:c6:44:0d:97:2d:13:
                b7:e3:bf:4e:be:b5:6a:df:ce:c4:c3:a4:8a:1d:fa:cf:
                99:dc:4a:17:61:e0:37:2b:7f:90:cb:31:02:97:e4:30:
                93:5d:91:f7:ef:b0:5a:c7:d4:de:d8:0e:b8:06:06:23:
                ed:5f:33:f3:f8:7e:09:c5:de:a5:32:2a:1b:6a:75:c5:
                0b:e3:a5:f2:7a:df:3e:3d:93:bf:ca:1f:d9:8d:24:ed
            Exponent: 65537 (0x10001)
Signature Algorithm: PKCS #1 MD5 With RSA Encryption
Signature:
    85:92:42:1e:e3:04:4d:e5:a8:79:12:7d:72:c0:bf:45:
    ea:c8:f8:af:f5:95:f0:f5:83:23:15:0b:02:73:82:24:
    3d:de:1e:95:04:fb:b5:08:17:04:1c:9d:9c:9b:bd:c7:
    e6:57:6c:64:38:8b:df:a2:67:f0:39:f9:70:e9:07:1f:
    33:48:ea:2c:18:1d:f0:30:d8:ca:e1:29:ec:be:a3:43:
    6f:df:03:d5:43:94:8f:ec:ea:9a:02:82:99:5a:54:c9:
    e4:1f:8c:ae:e2:e8:3d:50:20:46:e2:c8:44:a6:32:4e:
    51:48:15:d6:44:8c:e6:d2:0d:5f:77:9b:62:80:1e:30
Fingerprint (MD5):
    D9:FB:74:9F:C3:EC:5A:89:8F:2C:37:47:2F:1B:D8:8F
Fingerprint (SHA1):
    2E:CA:B8:BE:B6:A0:8C:84:0D:62:57:85:C6:73:14:DE:67:4E:09:56

Certificate Trust Flags:
    SSL Flags:
        Valid CA
        Trusted CA
        User
        Trusted Client CA
    Email Flags:
        User
    Object Signing Flags:
        User
```

▼ To Renew an Expired CA-Signed Server Certificate

When your CA-signed server certificate (public key and private key) expires, renew it by using this procedure.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Obtain an updated CA-signed server certificate from your Certificate Authority.**

2. **When you receive the updated certificate, stop the server instance and install the certificate.**

   ```
   $ dsadm stop instance-path
   $ dsadm renew-cert instance-path cert-alias cert-file
   ```

3. **Restart the server instance.**

   ```
   $ dsadm start instance-path
   ```

To Export and Import a CA-Signed Server Certificate

In some cases you might want to export the public and private keys of a certificate so that you can later import the certificate. For example, you might want the certificate to be used by another server.

The commands in this procedure can be used with certificates that contain wild cards, for example "`cn=*,o=example`".

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Export the certificate.**

   ```
   $ dsadm export-cert [-o output-file] instance-path cert-alias
   ```

   For example:

   ```
   $ dsadm export-cert -o /tmp/first-certificate /local/ds1 "First Certificate"
   $ dsadm export-cert -o /tmp/first-ca-server-certificate /local/ds1/ defaultCert

   Choose the PKCS#12 file password:
   Confirm the PKCS#12 file password:

   $ ls /tmp
   first-ca-server-certificate
   ```

2. **Import the certificate.**

   ```
   $ dsadm import-cert instance-path cert-file
   ```

   For example, to import the certificate to a server instance:

   ```
   $ dsadm import-cert /local/ds2 /tmp/first-ca-server-certificate
   Enter the PKCS#12 file password:
   ```

3 **(Optional) If you have imported the certificate to a server, configure the server to use the imported certificate.**

```
$ dsconf set-server-prop -e -h host -p port ssl-rsa-cert-name:server-cert
```

Configuring the Certificate Database Password

By default, Directory Server manages the SSL certificate database password internally through a stored password. When managing certificates, the user does not need to type a certificate password or specify the password file. This option is not very secure because the password is only hidden, not encrypted.

However, if you want to have more control over the use of certificates, you can configure the server so that the user is prompted for a password on the command line. In this case, the user must type the certificate database password for all `dsadm` subcommands except `autostart`, `backup`, `disable-service`, `enable-service`, `info`, `reindex`, `restore`, and `stop`. The certificate database is located in the directory *instance-path*`/alias`.

▼ To Configure the Server So the User is Prompted for a Certificate Password

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Stop the server.**

```
$ dsadm stop instance-path
```

2 **Set the password prompt flag to** `on`**.**

```
$ dsadm set-flags instance-path cert-pwd-prompt=on
```

You are asked to choose a new certificate password.

3 **Start the server.**

```
$ dsadm start instance-path
```

Backing Up and Restoring the Certificate Database for Directory Server

When you back up an instance of Directory Server, you back up the Directory Server configuration and the certificates. The backed up certificates are stored in the *archive-path*`/alias` directory.

For information about how to back up and restore Directory Server, see "To Make a Backup for Disaster Recovery" on page 231.

Configuring SSL Communication

This section contains procedures that help you to choose encryption ciphers.

Disabling Non Secure Communication

When a server instance is created, both an LDAP clear port and a secure LDAP port (LDAPS) are created by default. However, there might be situations where you want to disable non-SSL communications so that the server communicates only through SSL.

The SSL connection is enabled with a default self-signed certificate. If you want to, you can install your own certificates. For instructions on managing certificates and disabling SSL after the server has been started, see Chapter 5, "Directory Server Security." For an overview of certificates, certificate databases, and obtaining a CA-signed server certificate, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

▼ To Disable the LDAP Clear Port

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Disable the LDAP clear port.

To disable the non secure point, you must bind to the LDAP secure port. This example shows a bind to the default LDAP secure port, `1636`, on the host server `host1`.

```
$ dsconf set-server-prop -h host1 -P 1636 ldap-port:disabled
```

2 Restart the server for the change to take effect.

```
$ dsadm restart /local/dsInst
```

You can now no longer bind on the non secure port 1389.

Choosing Encryption Ciphers

A *cipher* is the algorithm used to encrypt and decrypt data. Generally speaking, the more bits that a cipher uses during encryption, the *stronger* or more secure the encryption is. Ciphers for SSL are also identified by the type of message authentication used. Message authentication is another algorithm that computes a `checksum` that guarantees data integrity.

When a client initiates an SSL connection with a server, the client and server must agree on a cipher to use to encrypt information. In any two-way encryption process, both parties must use the same cipher. The cipher used depends upon the current order of the cipher list kept by the server. The server chooses the first cipher presented by the client that matches a cipher in its list.

The default cipher value for Directory Server is `all`, which means all known secure ciphers supported by the underlying SSL library. However, you can modify this value to only accept certain ciphers.

For more information about the ciphers that are available with Directory Server, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

▼ To Choose an Encryption Cipher

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Make sure that SSL is enabled for your server.

See "Configuring SSL Communication" on page 111.

2 View the available SSL ciphers.

```
$ dsconf get-server-prop -h host -p port ssl-supported-ciphers
ssl-supported-ciphers  :  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_DSS_WITH_AES_256_CBC_SHA
...
```

3 (Optional) If you want to keep a copy of non-encrypted data, export the data before setting the SSL ciphers.

See "Exporting to LDIF" on page 225.

4 Set the SSL ciphers.

```
$ dsconf set-server-prop -h host -p port ssl-cipher-family:cipher
```

For example, to set the cipher family to `SSL_RSA_WITH_RC4_128_MD5` and `SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA`, type:

```
$ dsconf set-server-prop -h host1 -P 1636 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Enter "cn=Directory Manager" password:
Before setting SSL configuration, export Directory Server data.
Do you want to continue [y/n] ? y
Directory Server must be restarted for changes to take effect.
```

5 (Optional) Add an SSL cipher to an existing list.

If you already have a list of ciphers specified, and you want to add a cipher, use this command:

```
$ dsconf set-server-prop -h host -p port ssl-cipher-family+:cipher
```

For example, to add the `SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA` cipher, type:

```
$ dsconf set-server-prop -h host1 -P 1636 \
 ssl-cipher-family+:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
```

6 **Restart the server for the changes to take effect.**

```
$ dsadm restart /local/dsInst
```

Configuring Credential Levels and Authentication Methods

The security model that is applied to clients is defined through a combination of the credential level and the authentication method.

Directory Server supports the following credential levels:

- **Anonymous.** Allowing anonymous access for certain parts of the directory implies that anyone with access to the directory has read access.

  If you use an anonymous credential level, you need to allow read access to all the LDAP naming entries and attributes.

Caution – Do not allow anonymous write access to a directory because anyone could change information in the DIT to which they have write access, including another user's password, or their own identity.

- **Proxy.** The client authenticates or binds to the directory using a proxy account.

  This proxy account can be any entry that is allowed to bind to the directory. The proxy account needs sufficient access to perform the naming service functions on the directory. You need to configure the `proxyDN` and `proxyPassword` on every client using the proxy credential level. The encrypted `proxyPassword` is stored locally on the client.
- **Proxy anonymous.** A multi-valued entry in which more than one credential level is defined.

  A client assigned the proxy anonymous level will first attempt to authenticate with its proxy identity. If the client is unable to authenticate as the proxy user for whatever reason (user lockout, password expired, for example), then the client will use anonymous access. This might lead to a different level of service, depending on how the directory is configured.

Client authentication is a mechanism for the server to verify the identity of the client.

Client authentication can be performed in one of the following ways:

- By providing a DN and a password.
- Through a certificate presented by the client.

  Certificate-based authentication uses the client certificate that is obtained through the SSL protocol to find a user entry for identification. In certificate-based authentication, the client sends an SASL bind request specifying an external mechanism. The bind request relies on the SSL authentication mechanism that has already been established.
-

  Through a SASL-based mechanism.

  - On all operating systems, SASL through DIGEST-MD5.
  - On the Solaris Operating System, SASL through the GSSAPI mechanism which allows the authentication of a client through Kerberos V5.

  When using either of the two SASL mechanisms, the server must also be configured to perform identity mapping. The SASL credentials are called the *Principal*. Each mechanism must have specific mappings to determine the bind DN from the contents of the Principal. When the Principal is mapped to a single user entry and the SASL mechanism validates that user's identity, the user's DN is the bind DN for the connection.
- Through SSL client authentication mode.

  Use SSL client authentication when you want all clients to be authorized at the SSL layer. The client application authenticates by sending its SSL certificate to the server. You specify whether the server allows, requires, or does not allow SSL client authentication using the `SSL-client-auth-mode` flag. By default, clients are allowed to authenticate.

This section provides the following information about configuring the two SASL mechanisms on Directory Server.

- "Setting SASL Encryption Levels in Directory Server" on page 114
- "SASL Authentication Through DIGEST-MD5" on page 116
- "SASL Authentication Through GSSAPI (Solaris OS Only)" on page 118

For more information about configuring security, see "Configuring LDAP Clients to Use Security" on page 121.

Setting SASL Encryption Levels in Directory Server

Before configuring the SASL mechanism, you must specify whether you require encryption or not. Requirements for SASL encryption are set by the maximum and minimum Strength Security Factor (SSF).

The attributes `dsSaslMinSSF(5dsat)` and `dsSaslMaxSSF(5dsat)` represent the encryption key length, and they are stored in `cn=SASL, cn=security, cn=config`.

The server allows any level of encryption, including no encryption. This means that Directory Server accepts `dsSaslMinSSF` and `dsSaslMaxSSF` values greater than 256. However, no SASL mechanisms currently support an SSF greater than 128. Directory Server negotiates these values down to the highest SSF possible (128). Therefore, the highest actual SSF might be less than the configured maximum, depending on the underlying mechanisms available.

SASL security factor authentication depends two main items: the minimum and maximum factors requested by the server and client applications, and the available encryption mechanisms, which are provided by the underlying security components. In summary, the server and client attempt to use the highest available security factor that is less than or equal to the maximum factors set on both, but greater than or equal to the minimum factors on both.

The default minimum SASL security factor for Directory Server, `dsSaslMinSSF`, is `0`, meaning no protection. The actual minimum depends on the client setting, unless you change the minimum for Directory Server. In practice, you should set the minimum to the lowest level that you actually want the server and client to use. If the server and client fail to negotiate a mechanism that meets the minimum requirements, the connection is not established.

▼ To Require SASL Encryption

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **To require SASL encryption, set the `dsSaslMinSSF` value to the minimum encryption required.**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 128
^D
```

▼ To Disallow SASL Encryption

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **To disallow SASL encryption, set both the `dsSaslMinSSF` and `dsSaslMaxSSF` values to zero.**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 0

replace: dsSaslMaxSSF
dsSaslMaxSSF: 0
```

SASL Authentication Through DIGEST-MD5

The DIGEST-MD5 mechanism authenticates clients by comparing a hashed value sent by the client with a hash of the user's password. However, because the mechanism must read user passwords, all users that want to be authenticated through DIGEST-MD5 must have `{CLEAR}` passwords in the directory. When storing `{CLEAR}` passwords in the directory, you must ensure that access to password values is properly restricted through ACIs, as described in Chapter 6, "Directory Server Access Control." In addition, you need to configure attribute encryption in the suffix, as described in "Encrypting Attribute Values" on page 95.

▼ To Configure the DIGEST-MD5 Mechanism

The following procedure explains how to configure Directory Server to use DIGEST-MD5.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Use the `ldapsearch` command to verify that `DIGEST-MD5` is a value of the `supportedSASLMechanisms` attribute on the root entry.

For example, the following command shows which SASL mechanisms are enabled:

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -s base -b "" "(objectclass=*)" supportedSASLMechanisms
Enter bind password:
dn:
supportedSASLMechanisms: EXTERNAL
supportedSASLMechanisms: DIGEST-MD5
supportedSASLMechanisms: GSSAPI
```

2 If DIGEST-MD5 is not enabled, enable it.

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
add: dsSaslPluginsEnable
dsSaslPluginsEnable: DIGEST-MD5
-
replace: dsSaslPluginsPath
dsSaslPluginsPath: SASL-library
```

where *SASL-library* is one of the following:

Native packages based installation	`/usr/lib/mps/sasl2`
Zip installation	*install-path*`/lib/private/`

3 Use the default identity mapping for DIGEST-MD5, or create new ones.

For information, see "DIGEST-MD5 Identity Mappings" on page 117.

4 **Ensure that the password is stored in `{CLEAR}` for all users who will access the server through SSL using DIGEST-MD5.**

See Chapter 7, "Directory Server Password Policy," for password storage schemes.

5 **If you modified the SASL configuration entry or one of the DIGEST-MD5 identity mapping entries, restart Directory Server.**

DIGEST-MD5 Identity Mappings

Identity mappings for SASL mechanisms try to match the credentials of the SASL identity with a user entry in the directory. Authentication fails if the mapping cannot find a DN that corresponds to the SASL identity. See *Sun Directory Server Enterprise Edition 7.0 Reference* for a complete description of this mechanism.

The SASL identity is a string called the *Principal* that represents a user in a format specific to each mechanism. In DIGEST-MD5, clients should create a Principal that contains either a `dn:` prefix and an LDAP DN or a `u:` prefix followed by any text determined by the client. During the mapping, the Principal that is sent by the client is available in the `${Principal}` placeholder.

The following entry in your server configuration is the default identity mapping for DIGEST-MD5:

```
dn: cn=default,cn=DIGEST-MD5,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: \${Principal}
dsMatching-regexp: dn:(.*)
dsMappedDN: \$1
```

This identity mapping assumes that the `dn` field of the Principal contains the exact DN of an existing user in the directory.

▼ To Define Your Own Identity Mappings for DIGEST-MD5

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Edit the default mapping entry or create new mapping entries under `cn=DIGEST-MD5,cn=identity mapping,cn=config`.**

The following command shows how this mapping would be defined:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
```

```
dn: cn=unqualified-username,cn=DIGEST-MD5,cn=identity mapping
cn=config
objectclass: dsIdentityMapping
objectclass: dsPatternMatching
objectclass: nsContainer
objectclass: top
cn: unqualified-username
dsMatching-pattern: \${Principal}
dsMatching-regexp: u:(.*)@(.*)\\.com
dsSearchBaseDN: dc=\$2
dsSearchFilter: (uid=\$1)
```

2 **Restart Directory Server for your new mappings to take effect.**

SASL Authentication Through GSSAPI (Solaris OS Only)

The Generic Security Service API (GSSAPI) over SASL allows you to use a third-party security system such as Kerberos V5 to authenticate clients. The GSSAPI library is available only for the Solaris OS SPARC® platform. Sun recommends that you install the Kerberos V5 implementation on the Sun Enterprise Authentication Mechanism™ 1.0.1 server.

The server uses the GSSAPI to validate the identity of the user. Then, the SASL mechanism applies the GSSAPI mapping rules to obtain a DN that is the bind DN for all operations during this connection.

▼ To Configure the Kerberos System

Configure the Kerberos software according to the manufacturer's instructions. If you are using the Sun Enterprise Authentication Mechanism 1.0.1 server, use this procedure.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Configure the files in `/etc/krb5`.**

2 **Create the Kerberos database for storing users and services.**

3 **In the database, create the Principal for the LDAP service.**

```
$ ldap/server-FQDN@realm
```

where *server-FQDN* is the fully qualified domain name of your Directory Server.

4 **Start the Kerberos daemon processes.**

Note – The DNS must be configured on the host machine.

Refer to your software documentation for detailed instructions for each of these steps. Also, see "Example Configuration of Kerberos Authentication Using GSSAPI With SASL" on page 124.

▼ To Configure the GSSAPI Mechanism

The following procedure explains how to configure Directory Server to use GSSAPI on the Solaris OS:

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Create the default identity mapping for GSSAPI and any custom mappings as described in "GSSAPI Identity Mappings" on page 119.**

2. **Create a keytab to store the service keys.**

   Your LDAP service key is stored in the keytab.

   a. **Ensure that the keytab is only readable by the Directory Server user.**

   b. **Change the file name to be different from the default `/etc/krb5/krb5.keytab`.**

   c. **Set the environment variable `KRB5_KTNAME` to ensure that the new keytab is used rather than the default keytab.**

3. **If you modified the SASL configuration entry or one of the GSSAPI identity mapping entries, restart Directory Server.**

   Note that the DNS must be configured on the host machine.

GSSAPI Identity Mappings

Identity mappings for SASL mechanisms try to match credentials of the SASL identity with a user entry in the directory. Authentication fails if the mapping cannot find a DN that corresponds to the SASL identity.

The SASL identity is a string called the *Principal* that represents a user in a format specific to each mechanism. In Kerberos using GSSAPI, the Principal is an identity with the format *uid* `[/`*instance*`][@`*realm*`]`. The *uid* can contain an optional *instance* identifier followed by an optional *realm* that is often a domain name. For example, the following strings are all valid user Principals:

```
bjensen
bjensen/Sales
bjensen@EXAMPLE.COM
bjensen/Sales@EXAMPLE.COM
```

Initially, no GSSAPI mappings are defined in the directory. Define a default mapping and any custom mappings that you need according to how your clients define the Principal that your clients use.

▼ To Define Identity Mappings for GSSAPI

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Create new mapping entries under** `cn=GSSAPI,cn=identity mapping, cn=config`**.**

See *Sun Directory Server Enterprise Edition 7.0 Reference* for the definition of the attributes in an identity mapping entry. Examples of GSSAPI mappings are located in *instance-path*`/ldif/identityMapping_Examples.ldif`.

The default GSSAPI mapping in this file assumes that the Principal contains only a user ID. This mapping determines a user in a fixed branch of the directory:

```
dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
objectclass: dsIdentityMapping
objectclass: nsContainer
objectclass: top
cn: default
dsMappedDN: uid=\${Principal},ou=people,dc=example,dc=com
```

Another example in this file shows how to determine the user ID when the user ID is contained in a Principal that includes a known realm.

```
dn: cn=same_realm,cn=GSSAPI,cn=identity mapping,cn=config
objectclass: dsIdentityMapping
objectclass: dsPatternMatching
objectclass: nsContainer
objectclass: top
cn: same_realm
dsMatching-pattern: \${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=\$1,ou=people,dc=EXAMPLE,dc=COM
```

2 **Restart Directory Server for your new mappings to take effect.**

Configuring LDAP Clients to Use Security

The following sections explain how to configure and use SSL in LDAP clients that want to establish secure connections with Directory Server. In an SSL connection, the server sends its certificate to the client. The client must first authenticate the server by trusting its certificate. Then, the client can optionally initiate one of the client authentication mechanisms by sending its own certificate or information for one of the two SASL mechanism. The SASL mechanisms are DIGEST-MD5 and GSSAPI using Kerberos V5.

The following sections use the `ldapsearch` tool as an example of an SSL-enabled LDAP client.

To configure SSL connections on other LDAP clients, refer to the documentation provided with your application.

Note – Some client applications implement SSL but do not verify that the server has a trusted certificate. These client applications use the SSL protocol to provide data encryption but cannot guarantee confidentiality nor protect against impersonation.

The following sections explain how to configure LDAP clients to use security:

Using SASL DIGEST-MD5 in Clients

When using the DIGEST-MD5 mechanism in clients, you do not need to install a user certificate. However, if you want to use encrypted SSL connections, you must still trust the server certificate as described in "Managing Certificates" on page 103.

Specifying a Realm

A *realm* defines the namespace from which the authentication identity is selected. In DIGEST-MD5 authentication, you must authenticate to a specific realm.

Directory Server uses the fully qualified host name of the machine as the default realm for DIGEST-MD5. The server uses the lowercase value of the host name that is found in the `nsslapd-localhost` configuration attribute.

If you do not specify a realm, the default realm offered by the server is used.

Specifying Environment Variables

In the UNIX environment, you must set the `SASL-PATH` environment variable so that the LDAP tools can find the DIGEST-MD5 libraries. The DIGEST-MD5 library is a shared library that is dynamically loaded by the SASL plug-in. Set the `SASL_PATH` environment variable as follows:

```
export SASL_PATH=SASL-library
```

This path assumes that Directory Server is installed on the same host where the LDAP tools are invoked.

Examples of the `ldapsearch` Command

You can perform DIGEST-MD5 client authentication without using SSL. The following example uses the default DIGEST-MD5 identity mapping to determine the bind DN:

```
$ ldapsearch -h host1 -p 1389 \
 -o mech=DIGEST-MD5 [ \
 -o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -w - \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
 -o secProp="minssf=56,maxssf=256,noplain" \
 -b "dc=example,dc=com" "(givenname=Richard)"
```

The preceding example shows the use of the `-o` (lowercase letter o) option to specify SASL options. The `realm` is optional, but if specified, it must be the fully qualified domain name of the server host machine. The `authid` and `authzid` must both be present and identical, although the `authzid` intended for proxy operations is not used. The `-w` password option applies to the `authid`.

The value of `authid` is the Principal used in identity mapping. The `authid` should contain either the `dn:` prefix followed by a valid user DN in the directory, or the `u:` prefix followed by any string determined by the client. This use of `authid` allows you to use the mappings that are shown in "DIGEST-MD5 Identity Mappings" on page 117.

The most common configuration is for an SSL connection to provide encryption over the LDAPS secure port and DIGEST-MD5 to provide the client authentication. The following example performs the same operation over SSL:

```
$ ldapsearch -h host1 -P 1636 \
 -Z -P .mozilla/bjensen/BJE6001.slt/cert8.db \
 -N "cert-example" -w - \
 -o mech=DIGEST-MD5 [-o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
 -o secProp="minssf=0,maxssf=0,noplain" \
 -b "dc=example,dc=com" "(givenname=Richard)"
```

In this example, the `-N` and `-w` options are required by the `ldapsearch` command, as the operation is performed over SSL. However , these options are not used for client authentication. Instead, the server performs another DIGEST-MD5 identity mapping of the Principal in the `authid` value.

Using Kerberos SASL GSSAPI in Clients

When using the GSSAPI mechanism in clients, you do not need to install a user certificate, but you must configure the Kerberos V5 security system. Also, if you want to use encrypted SSL connections, you must trust the server certificate as described in "Managing Certificates" on page 103.

▼ To Configure Kerberos V5 on a Host

You must configure Kerberos V5 on the host machine where your LDAP clients will run.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Install Kerberos V5 according to its installation instructions.**

   Sun recommends installing the Sun Enterprise Authentication Mechanism 1.0.1 client software.

2. **Configure the Kerberos software.**

   Using the Sun Enterprise Authentication Mechanism software, configure the files under `/etc/krb5`. This configuration sets up the `kdc` server, and defines the default realm and any other configuration required by your Kerberos system.

3. **If necessary, modify the file `/etc/gss/mech` so that the first value that is listed is `kerberos_v5`.**

▼ To Specify SASL Options for Kerberos Authentication

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Before using a client application that is enabled with the GSSAPI mechanism, initialize the Kerberos security system with your user Principal.**

   ```
   $ kinit user-principal
   ```

   where the *user-principal* is your SASL identity, for example, `bjensen@example.com`.

2 **Specify SASL options for using Kerberos.**

Note that in the UNIX environment, you must set the `SASL_PATH` environment variable to the correct path for the SASL libraries. For example in the Korn shell:

```
$ export SASL_PATH=SASL-library
```

This path assumes that Directory Server is installed on the same host where the LDAP tools are invoked.

The following example of the `ldapsearch` tool shows the use of the `-o` (lowercase letter o) option to specify SASL options for using Kerberos:

```
$ ldapsearch -h www.host1.com -p 1389 -o mech=GSSAPI -o authid="bjensen@EXAMPLE.COM" \
 -o authzid="bjensen@EXAMPLE.COM" -b "dc=example,dc=com" "(givenname=Richard)"
```

The `authid` can be omitted because it is present in the Kerberos cache that was initialized by the `kinit` command. If `authid` is present, `authid` and `authzid` must be identical, although the `authzid` intended for proxy operations is not used. The value of `authid` is the Principal that is used in identity mapping. The Principal must be the full Principal, including the realm. See "GSSAPI Identity Mappings" on page 119.

Example Configuration of Kerberos Authentication Using GSSAPI With SASL

Configuring Kerberos for Directory Server can be complicated. Your first point of reference should be the Kerberos documentation.

For more help, use the following example procedure to get an idea of which steps to follow. Be aware, however, that this procedure is an example. You must modify the procedure to suit your own configuration and your own environment.

Additional information about configuring and using Kerberos in the Solaris OS can be found in *System Administration Guide: Security Services*. This guide is a part of the Solaris documentation set. You can also consult the man pages.

Information about this example and the steps used are as follows:

Assumptions for This Example

This example procedure describes the process of configuring one machine to operate as a Key Distribution Center (KDC), and a second machine to run a Directory Server. The result of this procedure is that users can perform Kerberos authentication through GSSAPI.

It is possible to run both the KDC and the Directory Server on the same machine. If you choose to run both on the same machine, use the same procedure, but omit the steps for the Directory Server machine that have already been done for the KDC machine.

This procedure makes a number of assumptions about the environment that is used. When using the example procedure, modify the values accordingly to suit your environment. These assumptions are:

- This system has a fresh installation of the Solaris 9 software with the latest recommended patch cluster installed. Kerberos authentication to the Directory Server can fail if the appropriate Solaris patches are not installed.

  Note that although the documented procedure is largely the same for Solaris 10, there are some differences. The configuration file format is slightly different, and the output of some of the commands might not be the same.
- The machine that is running the Kerberos daemons has the fully qualified domain name of `kdc.example.com`. The machine must be configured to use DNS as a naming service. This configuration is a requirement of Kerberos. Certain operations might fail if other naming services such as `file` are used instead.
- The machine that is running Directory Server has the fully qualified domain name of `directory.example.com`. This machine must also be configured to use DNS as a naming service.
- The Directory Server machine serves as the client system for authenticating to the Directory Server through Kerberos. This authentication can be performed from any system that can communicate with both the Directory Server and Kerberos daemons. However, all of the necessary components for this example are provided with the Directory Server, and the authentication is performed from that system.

- Users in the Directory Server have DNs of the form `uid=`*username*`,ou=People,dc=example,dc=com`. The corresponding Kerberos principal is *username*`@EXAMPLE.COM`. If a different naming scheme is used, a different GSSAPI identity mapping must be used.

All Machines: Edit the Kerberos Client Configuration File

The `/etc/krb5/krb5.conf` configuration file provides information that Kerberos clients require in order to communicate with the KDC.

Edit the `/etc/krb5/krb5.conf` configuration file on the KDC machine, the Directory Server machine, and any client machines that will authenticate to the Directory Server using Kerberos.

- Replace every occurrence of "`___default_realm___`" with "`EXAMPLE.COM`".
- Replace every occurrence of "`___master_kdc___`" with "`kdc.example.com`".
- Remove the line that contains "`___slave_kdcs___`" as there will be only a single Kerberos server.
- Replace "`___domain_mapping___`" with "`.example.com = EXAMPLE.COM`" (note the initial period in `.example.com`).

The updated `/etc/krb5/krb5.conf` configuration file should look like the contents of the following example.

EXAMPLE 5–1 Edited Kerberos Client Configuration File `/etc/krb5/krb5.conf`

```
#pragma ident   "@(#)krb5.conf  1.2     99/07/20 SMI"
# Copyright (c) 1999, by Sun Microsystems, Inc.
# All rights reserved.
#
# krb5.conf template
# In order to complete this configuration file
# you will need to replace the __<name\>__ placeholders
# with appropriate values for your network.
#

[libdefaults]
        default_realm = EXAMPLE.COM
[realms]
        EXAMPLE.COM = {
                kdc = kdc.example.com
                admin_server = kdc.example.com
        }
[domain_realm]
        .example.com = EXAMPLE.COM
[logging]
```

EXAMPLE 5–1 Edited Kerberos Client Configuration File `/etc/krb5/krb5.conf` *(Continued)*

```
        default = FILE:/var/krb5/kdc.log
        kdc = FILE:/var/krb5/kdc.log
        kdc_rotate = {

# How often to rotate kdc.log. Logs will get rotated no more
# often than the period, and less often if the KDC is not used
# frequently.
                period = 1d

# how many versions of kdc.log to keep around (kdc.log.0, kdc.log.1, ...)
                versions = 10
        }

[appdefaults]
        kinit = {
                renewable = true
                forwardable= true
        }
        gkadmin = {
                help_url =
 http://docs.sun.com:80/ab2/coll.384.1/SEAM/@AB2PageView/1195
        }
```

All Machines: Edit the Administration Server ACL Configuration File

Replace "`___default_realm___`" with "`EXAMPLE.COM`" in the `/etc/krb5/kadm5.acl` configuration file. The updated file should look like the following example.

EXAMPLE 5–2 Edited Administration Server ACL Configuration File

```
#
# Copyright (c) 1998-2000 by Sun Microsystems, Inc.
# All rights reserved.
#
# pragma ident   "@(#)kadm5.acl  1.1     01/03/19 SMI"
*/admin@EXAMPLE.COM *
```

KDC Machine: Edit the KDC Server Configuration File

Edit the `/etc/krb5/kdc.conf` file to replace "`___default_realm___`" with "`EXAMPLE.COM`". The updated file should look like the following example.

EXAMPLE 5–3 Edited KDC Server Configuration File `/etc/krb5/kdc.conf`

```
# Copyright 1998-2002 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)kdc.conf   1.2     02/02/14 SMI"

[kdcdefaults]
        kdc_ports = 88,750

[realms]
        EXAMPLE.COM = {
                profile = /etc/krb5/krb5.conf
                database_name = /var/krb5/principal
                admin_keytab = /etc/krb5/kadm5.keytab
                acl_file = /etc/krb5/kadm5.acl
                kadmind_port = 749
                max_life = 8h 0m 0s
                max_renewable_life = 7d 0h 0m 0s
                default_principal_flags = +preauth
        }
```

KDC Machine: Create the KDC Database

```
$ /usr/sbin/kdb5_util create -r EXAMPLE.COM -s

Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: password
Re-enter KDC database master key to verify: password
$
```

KDC Machine: Create an Administration Principal and Keytab

Use the following command to create an administration user with a Principal of `kws/admin@EXAMPLE.COM` and service keys that will be used by the administration daemon.

```
$ /usr/sbin/kadmin.local
kadmin.local:  add_principal kws/admin
Enter password for principal "kws/admin@EXAMPLE.COM": secret
Re-enter password for principal "kws/admin@EXAMPLE.COM": secret
Principal "kws/admin@EXAMPLE.COM" created.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/kdc.example.com
Entry for principal kadmin/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
```

```
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab changepw/kdc.example.com

Entry for principal changepw/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/changepw
Entry for principal kadmin/changepw with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  quit$
```

KDC Machine: Start the Kerberos Daemons

Run the following commands to start the KDC and administration daemons:

```
$ /etc/init.d/kdc start
$ /etc/init.d/kdc.master start
$
```

The KDC process will appear in the process list as `/usr/lib/krb5/krb5kdc`. The administration daemon will appear as `/usr/lib/krb5/kadmind`.

Note than in the Solaris 10 OS, the daemons are managed by the Service Management Facility (SMF) framework. Start the daemons on the Solaris 10 OS:

```
$ svcadm disable network/security/krb5kdc
$ svcadm enable network/security/krb5kdc
$ svcadm disable network/security/kadmin
$ svcadm enable network/security/kadmin
$
```

KDC Machine: Add Host Principals for the KDC and Directory Server Machines

Use the following sequence of commands to add host Principals to the Kerberos database for the KDC and Directory Server machines. The host Principal is used by certain Kerberos utilities such as `klist`.

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal -randkey host/kdc.example.com
Principal "host/kdc.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/kdc.example.com
Entry for principal host/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  add_principal -randkey host/directory.example.com
Principal "host/directory.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/directory.example.com
Entry for principal host/directory.example.com with kvno 3, encryption type
```

```
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  quit
$
```

KDC Machine: Add an LDAP Principal for the Directory Server

For the Directory Server to be able to validate the Kerberos tickets that are held by authenticating users, the Directory Server must have its own Principal. Currently, the Directory Server is hard coded to require a Principal of `ldap/`*fqdn*`@`*realm* where *fqdn* is the fully-qualified domain name of the Directory Server and *realm* is the Kerberos realm. The *fqdn* must match the fully qualified name provided when installing the Directory Server. In this case, the Principal for the Directory Server would be `ldap/directory.example.com@EXAMPLE.COM`.

Use the following sequence of commands to create an LDAP Principal for the Directory Server:

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin: add_principal -randkey ldap/directory.example.com
Principal "ldap/directory.example.com@EXAMPLE.COM" created.
kadmin: quit
$
```

KDC Machine: Add a Test User to the KDC

To perform Kerberos authentication, the user authenticating must exist in the Kerberos database. In this example, the user has the user name `kerberos-test`, which means that the Kerberos Principal is `kerberos-test@EXAMPLE.COM`.

Create the user by using the command sequence in this example:

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal kerberos-test
Enter password for principal "kerberos-test@EXAMPLE.COM": secret

Re-enter password for principal "kerberos-test@EXAMPLE.COM": secret

Principal "kerberos-test@EXAMPLE.COM" created.
kadmin:  quit
$
```

Directory Server Machine: Install the Directory Server

Install Directory Server 6.0 and the latest patches. Following are example settings.

Variable Type	Example Value
Fully qualified computer name	`directory.example.com`
Installation directory	`/opt/SUNWdsee`
Instance path	`/local/dsInst`
Server user	`unixuser`
Server group	`unixgroup`
Server port	`389`
Suffix	`dc=example,dc=com`

Directory Server Machine: Configure the Directory Server to Enable GSSAPI

First, create the file `/data/ds/shared/bin/gssapi.ldif` to define the mapping that should be used by the Directory Server, and to identify which Kerberos user is authenticating, based on the Principal. Create the file contents to be the same as what is shown in the following example.

EXAMPLE 5–4 `gssapi.ldif` File Contents

```
dn: cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
objectClass: top
objectClass: nsContainer
cn: GSSAPI
dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: \${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=\$1,ou=People,dc=example,dc=com

dn: cn=SASL,cn=security,cn=config
changetype: modify
replace: dsSaslPluginsPath
dsSaslPluginsPath: /usr/lib/mps/sasl2/libsasl.so
```

Next, use the `ldapmodify` command to update the Directory Server to enable GSSAPI with the appropriate mappings, as shown in the following example:

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -a \
-f /data/ds/shared/bin/gssapi.ldif
adding new entry cn=GSSAPI,cn=identity mapping,cn=config
adding new entry cn=default,cn=GSSAPI,cn=identity mapping,cn=config
modifying entry cn=SASL,cn=security,cn=config
```

Directory Server Machine: Create a Directory Server Keytab

As mentioned previously, to authenticate Kerberos users through GSSAPI, the Directory Server must have its own Principal in the KDC. For authentication to work properly, the Principal information must reside in a Kerberos keytab on the Directory Server machine. This information must be in a file that is readable by the user account under which the Directory Server operates.

Create a keytab file with the correct properties by using the following command sequence:

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  ktadd -k /local/dsInst/config/ldap.keytab ldap/directory.example.com
Entry for principal ldap/directory.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab
 WRFILE:/local/dsInst/config/ldap.keytab.
kadmin:  quit
$
```

Change the permissions and ownership on this custom keytab. Make the keytab owned by the user account used to run Directory Server and readable only by that user:

```
$ chown unixuser:unixgroup /local/dsInst/config /ldap.keytab
$ chmod 600 /local/dsInst/config/ldap.keytab
$
```

By default, the Directory Server tries to use the standard Kerberos keytab in the file `/etc/kerb5/krb5.keytab`. However, making this file readable by the Directory Server user could constitute a security risk, which is why a custom keytab was created for the Directory Server.

Configure the Directory Server to use the new custom keytab. Do this by setting the `KRB5_KTNAME` environment variable.

Finally, restart the Directory Server to allow these changes to take effect:

```
$ KRB5_KTNAME=/etc/krb5/ldap.keytab dsadm restart /local/dsInst
```

Directory Server Machine: Add a Test User to the Directory Server

To authenticate a Kerberos user to the Directory Server, there must be a directory entry for the user that corresponds to the Kerberos Principal for that user.

In a previous step, a test user was added to the Kerberos database with a Principal of `kerberos-test@EXAMPLE.COM`. Because of the identity mapping configuration added to the directory, the corresponding directory entry for that user must have a DN of `uid=kerberos-test,ou=People,dc=example,dc=com`.

Before you can add the user to the directory, you must create the file `testuser.ldif` with the following contents.

EXAMPLE 5–5 New `testuser.ldif` File

```
dn: uid=kerberos-test,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: kerberos-test
givenName: Kerberos
sn: Test
cn: Kerberos Test
description: An account for testing Kerberos authentication through GSSAPI
```

Next, use `ldapmodify` to add this entry to the server:

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f testuser.ldif
adding new entry uid=kerberos-test,ou=People,dc=example,dc=com
$
```

Directory Server Machine: Get a Kerberos Ticket as the Test User

The test user exists in the Kerberos database and Directory Server and the KDC. Therefore, it is now possible to authenticate as the test user to the Directory Server over Kerberos through GSSAPI.

First, use the `kinit` command to get a Kerberos ticket for the user, as shown in the following example:

```
$ kinit kerberos-test
Password for kerberos-test@EXAMPLE.COM: secret
$
```

Then, use the `klist` command to view information about this ticket:

```
$ klist
Ticket cache: /tmp/krb5cc_0
Default principal: kerberos-test@EXAMPLE.COM
```

```
Valid starting           Expires                  Service principal
Sat Jul 24 00:24:15 2004  Sat Jul 24 08:24:15 2004  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until Sat Jul 31 00:24:15 2004
$
```

Client Machine: Authenticate to the Directory Server Through GSSAPI

The final step is to authenticate to the Directory Server by using GSSAPI. The `ldapsearch` utility provided with the Directory Server provides support for SASL authentication, including GSSAPI, DIGEST-MD5, and EXTERNAL mechanisms. However, to bind by using GSSAPI you must provide the client with the path to the SASL library. Provide the path by setting the `SASL_PATH` environment variable to the `lib/sasl` directory:

```
$ SASL_PATH=SASL-library
$ export SASL_PATH
$
```

To actually perform a Kerberos-based authentication to the Directory Server using `ldapsearch`, you must include the `-o mech=GSSAPI` and `-o authzid=`*principal* arguments.

You must also specify the fully qualified host name, shown here as `-h directory.example.com`, which must match the value of the `nsslapd-localhost` attribute on `cn=config` for the server. This use of the `-h` option is needed because the GSSAPI authentication process requires the host name provided by the client to match the host name provided by the server.

The following example retrieves the `dc=example,dc=com` entry while authenticated as the Kerberos test user account created previously:

```
$ ldapsearch -h directory.example.com -p 389 -o mech=GSSAPI \
 -o authzid="kerberos-test@EXAMPLE.COM" -b "dc=example,dc=com" -s base "(objectClass=*)"
version: 1
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain
```

Check the Directory Server access log to confirm that the authentication was processed as expected:

```
$ tail -12 /local/dsInst/logs/access

[24/Jul/2004:00:30:47 -0500] conn=0 op=-1 msgId=-1 - fd=23 slot=23 LDAP
        connection from 1.1.1.8 to 1.1.1.8
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - RESULT err=14 tag=97
```

```
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - RESULT err=14 tag=97
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - RESULT err=0 tag=97
     nentries=0 etime=0 dn="uid=kerberos-test,ou=people,dc=example,dc=com"
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - SRCH base="dc=example,dc=com"
     scope=0 filter="(objectClass=*)" attrs=ALL
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - RESULT err=0 tag=101 nentries=1
     etime=0
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=5 - UNBIND
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=-1 - closing - U1
[24/Jul/2004:00:30:48 -0500] conn=0 op=-1 msgId=-1 - closed.
$
```

This example shows that the bind is a three-step process. The first two steps return LDAP result `14` (SASL bind in progress), and the third step shows that the bind was successful. The `method=sasl` and `mech=GSSAPI` tags show that the bind used the GSSAPI SASL mechanism. The `dn="uid=kerberos-test,ou=people,dc=example,dc=com"` at the end of the successful bind response shows that the bind was performed as the appropriate user.

Pass-Through Authentication

Pass-through authentication (PTA) is a mechanism by which bind requests are filtered by bind DN. One Directory Server (the delegator) receives the bind request and, based on the filter, can consult another Directory Server (the delegate) to authenticate bind requests. As part of this functionality, the PTA plug-in enables the delegator Directory Server to accept simple password-based bind operations for entries that are not necessarily stored in its local database.

- "PTA Plug-In and DSCC" on page 135
- "Configuring the PTA Plug-in" on page 136

PTA Plug-In and DSCC

The PTA plug-in is also used by DSCC for private communication with the server. When a server instance is registered in DSCC, the PTA plug-in is enabled and the DSCC URL is added as an argument.

```
$ dsconf get-plugin-prop -h host -p port "Pass Through Authentication"

argument          :  ldap://dscc_host:3998/cn=dscc
```

```
depends-on-named  :
depends-on-type   :
desc              :  pass through authentication plugin
enabled           :  on
feature           :  passthruauth
init-func         :  passthruauth_init
lib-path          :  install-path/lib/passthru-plugin.so
type              :  preoperation
vendor            :  Sun Microsystems, Inc.
version           :  7.0
```

Note – If your server is registered in DSCC and you need to use PTA, you must preserve the following settings while modifying the PTA plug-in.

- Keep the `enabled` property `on`.
- Keep the DSCC URL in the argument, although you can add other values to the `argument` property.

If the PTA plug-in is disabled or the DSCC URL is removed from the argument, the server instance will appear as `inaccessible` in DSCC. If this happens, DSCC will automatically give you the option of resetting the PTA plug-in.

You can also fix this problem by unregistering and registering the Directory Server instance into DSCC. To perform these operations, you can use either DSCC or the `dsccreg remove-server` and `dsccreg add-server` commands. For more information about the `dsccreg` command, see `dsccreg(1M)`.

Configuring the PTA Plug-in

PTA plug-in configuration information is specified in the `cn=Pass Through Authentication,cn=plugins,cn=config` entry on the PTA server.

The PTA plug-in is a system plug-in, which is disabled by default. It can be enabled and setup using the `dsconf` command or using DSCC.

Setting up the PTA Plug-In

1. Run the following `dsconf` commands:

   ```
   $ dsconf enable-plugin -h PTAhost -p port "Pass Through Authentication"
   $ dsconf set-plugin-prop -h PTAhost -p port "Pass Through Authentication" \
   argument:"ldap[s]://authenticatingHost[:port]/PTAsubtree options"
   ```

   The plug-in argument specifies the LDAP URL identifying the hostname of the authenticating directory server, an optional port, and the PTA subtree. If no port is specified, the default port is `389` with LDAP and `636` with LDAPS. You may also set the

optional connection parameters described in the following sections. If the *PTAsubtree* exists in the *PTAhost*, the plug-in will not pass the bind request to the *authenticatingHost*, and the bind will be processed locally without any pass-through.

2. Restart the server as described in "Starting, Stopping, and Restarting a Directory Server Instance" on page 45.

Configuring PTA to Use a Secure Connection

Because the PTA plug-in must send bind credentials including the password to the authenticating directory, we recommend using a secure connection. To configure the PTA directory to communicate with the authenticating directory over SSL:

- Configure and enable SSL in both the PTA and authenticating directories, as described in Chapter 5, "Directory Server Security."
- Create or modify the PTA plug-in configuration to use LDAPS and the secure port in the LDAP URL, for example:

  `ldaps://`*`host`*`:`*`secure-port`*`/`*`subtree`*

Setting the Optional Connection Parameters

The PTA plug-in arguments accept a set of optional connection parameters after the LDAP URL:

```
http[s]://host:port/subtree [maxconns,maxops,timeout,ldapver,connlife]
```

The parameters must be given in the order shown. Although these parameters are optional, if you specify one of them, you must specify them all. If you do not want to customize all parameters, specify their default values given below. Make sure there is a space between the subtree parameter and the optional parameters.

You can configure the following optional parameters for each LDAP URL:

- `maxconns` - The maximum number of connections the PTA server can open simultaneously to the authenticating server. This parameter limits the number of simultaneous binds that can be passed-through to the authenticating server. The default value is `3`.
- `maxops` - The maximum number of bind requests the PTA directory server can send simultaneously to the authenticating directory server within a single connection. This parameter further limits the number of simultaneous pass-through authentications. The default is value is `5`.
- `timeout` - The maximum delay in seconds that you want the PTA server to wait for a response from the authenticating server. The default value is `300` seconds (five minutes).
- `ldapver` - The version of the LDAP protocol you want the PTA server to use when connecting to the authenticating server. The allowed values are `2` for LDAPv2 and `3` for LDAPv3. The default value is `3`.

- `connlife` - The time limit in seconds within which the PTA server will reuse a connection to the authenticating server. If a bind in the PTA subtree is requested by a client after this time has expired, the server closes the PTA connection and opens a new one. The server will not close the connection unless a bind request is initiated and the server determines the timeout has been exceeded. If you do not specify this option, or if only one authenticating server is listed in the LDAP URL, no time limit will be enforced. If two or more hosts are listed, the default is `300` seconds (five minutes).

Note – While setting the `argument` property using the `dsconf` command, put the value in double quotes to protect spaces. For example:

```
dsconf set-plugin-prop -h PTAhost -p port "Pass Through Authentication"\
 argument:"ldaps://eastbak.example.com/ou=East,ou=People,dc=example,dc=com\
 3,5,300,3,300"
```

Specifying Multiple Servers and Subtrees

You may configure the PTA plug-in with multiple arguments to specify multiple authenticating servers, multiple PTA subtrees, or both. Each argument contains one LDAP URL and may have its own set of connection options.

When there are multiple authenticating servers for the same PTA subtree, they act as failover servers. The plug-in will establish connections to them in the order listed whenever a PTA connection reaches the timeout limit. If all connections time out, the authentication fails.

When there are multiple PTA subtrees defined, the plug-in will pass-through the authentication request to the corresponding server according to the bind DN. The following example shows four PTA plug-in arguments that define two PTA subtrees, each with a failover server for authentication and server-specific connection parameters:

```
$ dsconf set-plugin-prop -h PTAhost -p port "Pass Through Authentication"\
 argument:"ldaps://configdir.example.com/o=example.com\
 10,10,60,3,300"
$ dsconf set-plugin-prop -h PTAhost -p port "Pass Through Authentication"\
 argument+:"ldaps://configbak.example.com/o=example.com\
 10,10,60,3,300"
$ dsconf set-plugin-prop -h PTAhost -p port "Pass Through Authentication"\
 argument+:"ldaps://east.example.com/ou=East,ou=People,dc=example,dc=com\
 10,10,60,3,300"
$ dsconf set-plugin-prop -h PTAhost -p port "Pass Through Authentication"\
 argument+:"ldaps://eastbak.example.com/ou=East,ou=People,dc=example,dc=com\
 10,10,60,3,300"
```

◆ ◆ ◆ CHAPTER 6

Directory Server Access Control

The control of access to your directory is an integral part of creating a secure directory. This chapter describes access control instructions (ACIs) that determine which permissions are granted to users who access the directory.

While you are in the planning phase of your directory deployment, define an access control strategy that serves your overall security policy. See the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide* for tips on planning an access control strategy.

For additional information about ACIs, including ACI syntax and bind rules, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Creating, Viewing, and Modifying ACIs

You can create ACIs either by using Directory Service Control Center (DSCC) or by using the command line. Whichever method you choose, it is often easier to view and copy an existing ACI value, rather than to create a new ACI from scratch.

You can view and modify the `aci` attribute values in DSCC. For information about how to modify ACIs through DSCC, see the DSCC online help.

▼ To Create, Modify, and Delete ACIs

To create ACIs by using the command line, you first create the ACIs in a file using LDIF statements. Then you add the ACIs to your directory tree by using the `ldapmodify` command.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Create the ACI in an LDIF file.**

```
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target)(version 3.0; acl "name";permission bindrules;)
```

This example shows how to add an ACI. To modify or delete the ACI, replace `add` with `replace` or `delete`.

For more examples of ACIs that are commonly used, see "Access Control Usage Examples" on page 163.

2 **Make the change using the LDIF file.**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w - -f ldif-file
```

▼ To View ACI Attribute Values

ACIs are stored as one or more values of the `aci` attribute of an entry. The `aci` attribute is a multi-valued operational attribute that can be read and modified by directory users. Therefore, the ACI attribute itself should be protected by ACIs. Administration users are usually given full access to the `aci` attribute.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **View the ACI attribute value of an entry by running the following `ldapsearch` command:**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b entryDN -s base "(objectclass=*)" aci
```

The result is LDIF text that you can copy to your new LDIF ACI definition for editing. Because the value of an ACI is a long string, the output from the `ldapsearch` operation is likely to be displayed over several lines. In addition, the first space is a continuation marker. If you want the LDIF output to not contain a continuation marker, use the `-T` option. Take the output format into account when copying and pasting the LDIF output.

Note – To view the permissions that an `aci` value grants and denies, see "Viewing Effective Rights" on page 176.

▼ To View ACIs at the Root Level

When you create a suffix, some default ACIs are created at the top or root level. These ACIs allow the default administration user `cn=admin,cn=Administrators,cn=config` to have the same access rights to directory data as the Directory Manager.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **View the default root level ACIs.**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "" -s base "(objectclass=*)" aci
```

ACI Syntax

The `aci` attribute has the following syntax:

```
aci: [list of (target)](version 3.0; acl "name";[list of "permission bindRules;"])
```

The following values are used in the ACI syntax:

target	Specifies the entry, attributes, or set of entries and attributes for which you want to control access. The *target* can be a distinguished name, one or more attributes, or a single LDAP filter. The *target* is optional. When a target is not specified, the ACI applies to the entry on which it is defined and its subtree. For information about targets, see "ACI Targets" on page 142.
version 3.0	A required string that identifies the ACI version.
name	A required string that identifies the ACI. Although there are no restrictions on the name, it is good practice to use unique, descriptive names for ACIs. Using unique names, will allow you to use `Get Effective Rights` to determine which ACI is in force.
permission	States what rights you are allowing or denying. For information about permissions, see "ACI Permissions" on page 147.
bindRules	Specifies the credentials and bind parameters that a user has to provide to be granted access. Bind rules can also be based on user membership, group membership, or connection properties of the client. For information about bind rules, see "ACI Bind Rules" on page 150.

The permission and bind rule portions of the ACI are set as a pair, also called an Access Control Rule (ACR). The specified permission to access the target is granted or denied depending on whether the accompanying bind rule is evaluated to be true or false.

Multiple targets and multiple permission-bind rule pairs can be used. This allows you to refine both the entry and attributes being targeted and efficiently set multiple access controls for a given target. The following example shows an ACI with multiple targets and multiple permission-bind rule pairs:

```
aci: (targetdefinition)...(targetdefinition)(version 3.0;acl "name";
permission bindRule; ...; permission bindRule;)
```

In the following example, the ACI states that `bjensen` has rights to modify all attributes in her own directory entry:

```
aci: (target="ldap:///uid=bjensen,dc=example,dc=com")
  (targetattr="*")(targetscope="subtree")(version 3.0; acl "example";
   allow (write) userdn="ldap:///self";)
```

The following sections describe the syntax of targets, permissions and bind rules.

ACI Targets

An ACI target statement specifies the entry, attributes, or set of entries and attributes for which you want to control access.

Target Syntax

An ACI target statement has this syntax:

(*keyword* = "*expression*")

The following values are used in the target.

keyword — Indicates the type of target.

expression — Identifies the target. The quotation marks ("") around *expression* are syntactically required, although the current implementation accepts expressions like `targetattr=*`.

`=`
`!=` — Indicates that the target is or is not the object specified in the expression.

The `!=` operator cannot be used with the `targettrfilters` keyword or the `targetscope` keyword.

Target Keywords

For a description of target keywords, see the following sections:

- "`target` Keyword" on page 143
- "`targetattr` Keyword" on page 144
- "`targetfilter` Keyword" on page 145
- "`targattrfilters` Keyword" on page 146
- "`targetscope` Keyword" on page 146

The following table lists the target keywords and their associated expressions.

TABLE 6–1 Target Keywords and Their Expressions

Keyword	Type of target	Expression
`target`	A directory entry or its subtree	ldap:///*distinguished_name*
`targetattr`	The attributes of an entry	*attribute*
`targetfilter`	A set of entries or attributes that match an LDAP filter	*LDAP_filter*
`targattrfilters`	An attribute value or combination of values that match an LDAP filter	*LDAP_operation*:*LDAP_filter*
`targetscope`	The scope of the target	`base`, `onelevel`, `subtree`

`target` Keyword

The `target` keyword specifies that an ACI is defined for a directory entry. The `target` keyword uses the following syntax:

```
(target = "distinguished_name")
```

or

```
(target != "distinguished_name")
```

The distinguished name must be in the subtree rooted at the entry where the ACI is defined. For example, the following target may be used in an ACI on `ou=People,dc=example,dc=com`:

```
(target = "ldap:///uid=bjensen,ou=People,dc=example,dc=com")
```

The DN of the entry must be a distinguished name in string representation (RFC 4514). Therefore, characters that are syntactically significant for a DN, such as commas, must be escaped with a single backslash (\).

Wildcards, shown as asterisk characters can be used in the expression for the `target` keyword. The asterisk matches an attribute value, a substring of a value, or a DN component. For example, all of the following expressions match `uid=bjensen,ou=people,dc=example,dc=com`.

- `target= "ldap:///uid=bj*,ou=people,dc=example,dc=com"`
- `target= "ldap:///uid=*,ou=people,dc=example,dc=com"`
- `target= "ldap:///*,ou=people,dc=example,dc=com"`
- `target= "ldap:///uid=bjensen,*,dc=com"`
- `target= "ldap:///uid=bjensen*"`

The following further examples show permitted uses of wildcards.

- `target="ldap:///uid=*,dc=example,dc=com"`

  This target matches every entry in the entire `example.com` tree that has the UID attribute in the entry's RDN.

- `target="ldap:///*Anderson,ou=People,dc=example,dc=com"`

  This target matches every entry in the `ou=People` branch whose RDN ends with Anderson, regardless of the naming attribute.

- `target="ldap:///uid=*,ou=*,dc=example,dc=com"`

  This target matches every entry in the `example.com` tree whose distinguished name contains the `uid` and `ou` attributes.

Other usage of wildcards to such as `target="ldap:///uid=bjensen,o*,dc=com"` might be accepted, but are deprecated.

`targetattr` Keyword

The `targetattr` keyword specifies that an ACI is defined for one or more attributes in the targeted entries. The `targetattr` keyword uses the following syntax:

`(targetattr = "`*attribute*`")`

or

`(targetattr != "`*attribute*`")`

If no `targetattr` keyword is present, no attributes are targeted. To target all attributes, the `targetattr` keyword must be `targetattr="*"`.

Targeted attributes do not need to exist on the target entry or its subtree, but the ACI applies whenever they do.

Targeted attributes do not need to be defined in the schema. The absence of schema checking makes it possible to implement an access control policy before importing data and its schema.

The `targetattr` keyword can be used for multiple attributes, by using this syntax:

```
(targetattr = "attribute1 || attribute2|| ... attributeN")
```

Note – If you configure attribute aliases, you must specify both the attribute name *and* the alias in the `targetattr` keyword for the ACI to take them into account.

Targeted attributes include all subtypes of the named attribute. For example, `(targetattr = "locality")` also targets `locality;lang-fr`.

Wildcards can be used in the expression for the `targetattr` keyword, but the use of wildcards would serve no purpose and may reduce performance.

`targetfilter` Keyword

The `targetfilter` keyword is used in ACIs to target entries that match an LDAP filter. The ACI applies to all entries that match the LDAP filter *and* that are in the scope of the ACI. The `targetfilter` keyword uses the following syntax:

```
(targetfilter = "(standard LDAP search filter)")
```

EXAMPLE 6–1 Using the `targetfilter` Keyword to Target Specific Entries

The following example is for employees with a status of salaried or contractor, and an attribute for the number of hours worked as a percentage of a full-time position. The filter targets entries for contractors or part-time employees:

```
(targetfilter = "(|(status=contractor)(fulltime<=79))")
```

The Netscape extended filter syntax is not supported in ACIs. For example, the following target filter is not valid:

```
(targetfilter = "(locality:fr:=<= Québec)")
```

The filter syntax that describes matching rules for internationalized values is supported. For example, the following target filter is valid:

```
(targetfilter = "(locality:2.16.840.1.113730.3.3.2.18.1.4:=Québec)")
```

The `targetfilter` keyword selects whole entries as targets of the ACI. The `targetfilter` keyword and the `targetattr` keyword can be used together to create ACIs that apply to a subset of attributes in the targeted entries.

`targattrfilters` Keyword

The `targattrfilters` keyword is used in ACIs to target specific attribute values by using LDAP filters. By using the `targattrfilters` keyword, you can grant or deny permissions on an attribute if that attribute's value meets the criteria defined in the ACI. An ACI that grants or denies access based on an attribute's value, is called a *value-based ACI*. The `targattrfilters` keyword uses this syntax:

```
(targattrfilters="add=attr1:F1 && attr2:F2... && attrn:Fn, \
 del=attr1:F1 && attr2:F2 ... && attrn:Fn")
```

where

`add` represents the operation of creating an attribute.

`del` represents the operation of deleting an attribute.

`attrn` represents the target attributes.

`Fn` represents filters that apply only to the associated attribute.

The following conditions must be met when filters apply to entries, and those entries are created, deleted or modified:

- When an entry is created or deleted, each instance of that attribute must satisfy the filter.
- When an entry is modified, if the operation adds an attribute, then the add filter that applies to that attribute must be satisfied; if the operation deletes an attribute, then the delete filter that applies to that attribute must be satisfied.
- If individual values of an attribute already present in the entry are replaced, then the add and delete filters must be satisfied.

EXAMPLE 6–2 Using the `targattrfilters` Keyword to Allow Users to Add Roles to Their Own Entries

The following ACI allows users to add any role to their own entry, except the `superAdmin` role. It also allows users to add a telephone number with a 123 prefix.

```
(targattrfilters="add=nsroleDN:(!(nsRoleDN=cn=superAdmin)) \
 && telephoneNumber:(telephoneNumber=123*)")
```

`targetscope` Keyword

The `targetscope` keyword is used in ACIs to specify the scope of the ACI. The `targetscope` keyword uses this syntax:

```
(targetscope="base")
```

The `targetscope` keyword can have one of these values:

`base`	The ACI applies to the target resource only
`onelevel`	The ACI applies to the target resource and its first-generation children
`subtree`	The ACI applies to the target resource and its subtree

If the `targetscope` keyword is not specified, the default value is `subtree`.

ACI Permissions

Permissions specify the type of access that is allowed or denied by the ACI. For information about bind rules, see the following sections:

- "Permission Syntax" on page 147
- "Permission Rights" on page 147
- "Permissions for Typical LDAP Operations" on page 148

Permission Syntax

An ACI permission statement has this syntax:

```
allow|deny (right1, right2 ...)
```

Rights define the operations you can perform on directory data. In an ACI statement, rights is a list of comma-separated keywords enclosed within parentheses.

Rights are granted independently of one another. This means, for example, that a user who is granted `add` rights but not `delete` rights can create an entry but cannot delete an entry. When you are planning the access control policy for your directory, ensure that you grant rights in a way that makes sense for users. For example, it might not make sense to grant `write` permission without granting `read` and `search` permissions.

Permission Rights

The following rights can be allowed or denied in an ACI permission statement:

Read	Permission to read directory data. This permission applies only to the search operation.
Write	Permission to modify an entry by adding, modifying, or deleting attributes. This permission applies to the modify and modify DN operations.
Add	Permission to create entries. This permission applies only to the add operation
Delete	Permission to delete entries. This permission applies only to the delete operation.

Search	Permission to search for directory data. Users must have `Search` and `Read` rights in order to view the data returned as part of a search result. This permission applies only to the search operation.
Compare	Permission for users to compare data they supply with data stored in the directory. With compare rights, the directory returns a success or failure message in response to an inquiry, but the user cannot see the value of the entry or attribute. This permission applies only to the compare operation.
Selfwrite	Permission for users to add or delete their own DN in an attribute of the target entry. The syntax of this attribute must be `distinguished name`. This right is used only for group management. The `Selfwrite` permission works with proxy authorization; it grants the right to add or delete the proxy DN from the group entry (not the DN of the bound user).
Proxy	Permission for the specified DN to access the target with the rights of another entry. You can grant proxy access using the DN of any user in the directory except the Directory Manager DN. You cannot grant proxy rights to the Directory Manager.
Import	Permission for an entry to be imported to the specified DN. This permission applies the modify DN operation.
Export	Permission for an entry to be exported from the specified DN. This permission applies the modify DN operation.
All	Permission for the specified DN to have the following rights for the targeted entry: `read`, `write`, `search`, `delete`, `compare`, and `selfwrite`. The `All` access right does control permission for the following rights to the target entry: `proxy`, `import`, and `export`.

Permissions for Typical LDAP Operations

This section describes the rights required to perform a set of LDAP operations.

Adding an entry:

- Grant add permission on the entry being added.
- Grant write permission on the value of each attribute in the entry. This right is granted by default but could be restricted using the `targettrfilters` keyword.

Deleting an entry:

- Grant delete permission on the entry to be deleted.
- Grant write permission on the value of each attribute in the entry. This right is granted by default but could be restricted using the `targettrfilters` keyword.

Modifying an attribute in an entry:

- Grant write permission on the attribute type.

- Grant write permission on the value of each attribute type. This right is granted by default but could be restricted using the `targettrfilters` keyword.

Modifying the RDN of an entry:

- Grant write permission on the entry.
- Grant write permission on the attribute type used in the new RDN.
- Grant write permission on the attribute type used in the old RDN, if you want to grant the right to delete the old RDN.
- Grant write permission on the value of attribute type used in the new RDN. This right is granted by default but could be restricted using the `targettrfilters` keyword.

Moving an entry to another subtree:

- Grant export permissions on the entry that you want to move.
- Grant import permission on the new superior entry of the entry that you want to move.

Comparing the value of an attribute:

Grant compare permission on the attribute type.

Searching for entries:

- Grant search permission on each attribute type used in the search filter.
- Grant read permission on at least one attribute type used in the entry to ensure that the entry is returned.
- Grant read permission an each attribute type to be returned with the entry.

EXAMPLE 6–3 Granting ACI Permissions to Perform a Search

The following ACI determines whether `bjensen` can be granted access for searching her own entry:

```
aci: (targetattr = "mail")(version 3.0; acl "self access to
 mail"; allow (read, search) userdn = "ldap:///self";)
```

The search result list is empty because this ACI does not allow `bjensen` the right to search on the `objectclass` attribute. To perform the search operation described, you must modify the ACI as follows:

```
aci: (targetattr = "mail || objectclass")(version 3.0; acl
 "self access to mail"; allow (read, search) userdn =
 "ldap:///self";)
```

ACI Bind Rules

Bind rules identify the set of users to which an ACI applies. The permission and bind rule portions of the ACI are set as a pair. The specified permission to access the target is granted or denied depending on whether the accompanying bind rule is evaluated to be true or false.

For information about bind rules, see the following sections:

Introduction to Bind Rules

Bind rules identify a set of users by using the following methods:

- The users, groups, and roles that are granted access.
- The location from which an entity must bind. The location from which a user authenticates can be spoofed and cannot be trusted. Do not base ACIs on this information alone.
- The time or day on which binding must occur.
- The type of authentication that must be in use during binding.

A simple bind rule might require a person accessing the directory to belong to a specific group. A complex bind rule can require a person to belong to a specific group and to log in from a machine with a specific IP address, between 8 am and 5 pm. Additionally, bind rules can be complex constructions that combine these criteria by using Boolean operators.

The server evaluates the logical expressions used in ACIs according to a three-valued logic, similar to the one used to evaluate LDAP filters, as described in section 4.5.1.7 of RFC 4511 *Lightweight Directory Access Protocol (v3)*. Therefore, if any component in the expression evaluates to Undefined (for example if the evaluation of the expression aborted due to a resource limitation), then the server handles this case correctly. The server does not erroneously grant access because an Undefined value occurred in a complex Boolean expression.

Bind Rule Syntax

An ACI bind rule has this syntax:

keyword = "*expression*";

or

keyword != "*expression*";

The following values are used in the bind rule:

keyword Indicates the type of bind rule.

expression Identifies the bind rule.

Bind Rule Keywords

For information about bind rule keywords, see the following sections:

The following table summarizes the keywords for bind rules.

TABLE 6–2 Bind Rule Keywords and Their Expressions

Keyword	Used to define access based on	Expression

`userdn`	Specified user	`ldap:///`*distinguished_name*
		`ldap:///all`
		`ldap:///anyone`
		`ldap:///self`
		`ldap:///parent`
		`ldap:///`*suffix*`??sub?(`*filter*`)`

`groupdn`	Specified group or groups	`[ldap:///`*DN*`]`
`roledn`	Specified role or roles	`[ldap:///`*DN*`]`
`userattr`	Matched attribute value	*attribute#bindType* or *attribute#value*
`ip`	Specified IP address or IP addresses	*IP_address*
`dns`	Specified domain or domains	*DNS_host_name*

TABLE 6–2 Bind Rule Keywords and Their Expressions *(Continued)*

Keyword	Used to define access based on	Expression
`timeofday`	Specified time of day	`0 - 2359`
`dayofweek`	Specified day or days of the week	`sun`, `mon`, `tue`, `wed`, `thu`, `fri`, and `sat`
`authmethod`	Specified authentication method	`none`, `simple`, `ssl`, `sasl` *authentication_method*

`userdn` Keyword

The `userdn` keyword is used to allow or deny access to a specified user. The following sections contain more information about the `userdn` keyword.

Syntax of the `userdn` Keyword

The `userdn` keyword uses this syntax:

```
userdn = "ldap:///dn [|| ldap:///dn]..."
userdn != "ldap:///dn [|| ldap:///dn]..."
```

The `userdn` keyword can alternatively be expressed as an LDAP URL filter. For information about expressing the `userdn` keyword as an LDAP URL, see "LDAP URLs in the `userdn` Keyword" on page 153.

dn can have of the following values:

distinguished-name — A fully qualified DN. Characters that are syntactically significant for a DN, such as commas, must be escaped with a single backslash (`\`). The wildcard * can be used to specify a set of users. For example, if the following user DN is specified, users with a bind DN beginning with the letter `b` are allowed or denied access:

```
uid=b*,dc=example,dc=com
```

`anyone` — Allows or denies access for anonymous and authenticated users, regardless of the circumstances of the bind.

This access can be limited to specific types of access (for example, access for read or access for search) or to specific subtrees or individual entries within the directory. The following ACI on the `dc=example,dc=com` node allows anonymous access to read and search the entire `dc=example,dc=com` tree.

```
aci: (version 3.0; acl "anonymous-read-search";
 allow (read, search) userdn = "ldap:///anyone";)
```

`all`
: Allows or denies access for authenticated users. This `all` value prevents anonymous access. The following ACI on the `dc=example,dc=com` node allows authenticated users to read the entire `dc=example,dc=com` tree:

```
aci: (version 3.0; acl "all-read";
 allow (read) userdn="ldap:///all";)
```

`self`
: Allows or denies users access to their own entries if the bind DN matches the DN of the targeted entry. The following ACI on the `dc=example,dc=com` node allows authenticated users in the `dc=example,dc=com` tree to write to their `userPassword` attribute.

```
aci: (targetattr = "userPassword")
 (version 3.0; acl "modify own password"; allow (write)
 userdn = "ldap:///self";)
```

`parent`
: Allows or denies users access to the entry if the bind DN is the parent of the targeted entry.

  The following ACI on the `dc=example,dc=com` node allows authenticated users in the `dc=example,dc=com` tree to modify any child entries of their bind DN:

```
aci: (version 3.0; acl "parent access";
 allow (write) userdn="ldap:///parent";)
```

LDAP URLs in the `userdn` Keyword

The `userdn` keyword can also be expressed as an LDAP URL with a filter, by using this syntax:

`userdn = ldap:///`*suffix*`??sub?(`*filter*`)`

LDAP URLs always apply to the local server. Do not specify a hostname or port number within an LDAP URL.

The following ACI on the `dc=example,dc=com` node allows all users in the `accounting` and `engineering` branches of the `example.com` tree to access to the targeted resource dynamically based on the following URL

```
userdn = "ldap:///dc=example,dc=com??sub?(|(ou=eng)(ou=acct))"
```

LDAP URLs can be used with the logical OR operator and the not-equal operator as shown in the following examples.

EXAMPLE 6–4 Using the `userdn` Keyword With a Logical OR Operator in LDAP URLs

This bind rule is evaluated to be true for users that bind with either of the specified DN patterns.

```
userdn = "ldap:///uid=b*,c=example.com || ldap:///cn=b*,dc=example,dc=com";
```

EXAMPLE 6–5 Using the `userdn` Keyword With a Not-Equal Operator in LDAP URLs

This bind rule is evaluated to be true if the client is not binding as a UID-based DN in the accounting subtree. This bind rule only makes sense if the targeted entry is not under the accounting branch of the directory tree.

```
userdn != "ldap:///uid=*,ou=Accounting,dc=example,dc=com";
```

`groupdn` Keyword

The `groupdn` keyword specifies that access to a targeted entry is granted or denied if the user binds by using a DN that belongs to a specific group. The `groupdn` keyword uses this syntax:

```
groupdn="ldap:///groupDN [|| ldap:///groupDN]..."
```

The bind rule is evaluated to be true if the bind DN belongs to a group that is specified by any of the values for *groupDN*.

In the following example, the bind rule is true if the bind DN belongs to the Administrators group :

```
aci: (version 3.0; acl "Administrators-write"; allow (write)
 groupdn="ldap:///cn=Administrators,dc=example,dc=com";)
```

Characters that are syntactically significant for a DN, such as commas, must be escaped with a single backslash (\).

`roledn` Keyword

The `roledn` keyword specifies that access to a targeted entry is granted or denied if the user binds using a DN that belongs to a specific role. The `roledn` keyword requires one or more valid distinguished names, in this format:

```
roledn = "ldap:///dn [|| ldap:///dn]... [|| ldap:///dn]"
```

The bind rule is evaluated to be true if the bind DN belongs to the specified role.

Characters that are syntactically significant for a DN, such as commas, must be escaped with a single backslash (\).

The `roledn` keyword has the same syntax and is used in the same way as the `groupdn` keyword.

`userattr` Keyword

The `userattr` keyword specifies which attribute values in the entry that was used to bind must match those in the targeted entry. The `userattr` keyword can be used for the following attributes:

- User DN
- Group DN
- Role DN
- LDAP filter, in an LDAP URL
- Any attribute type

An attribute generated by a Class of Service (CoS) definition cannot be used with the `userattr` keyword. ACIs that contain bind rules that depend on attribute values generated by CoS will not work.

The `userattr` keyword uses this syntax:

```
userattr = "attrName#bindType"
```

Alternatively, if you are using an attribute type that requires a value other than a user DN, group DN, role DN, or an LDAP filter, the `userattr` keyword uses this syntax:

```
userattr = "attrName#attrValue"
```

The `userattr` keyword can have one of the following values:

attrName The name of the attribute used for value matching

bindType One of the following types of bind: `USERDN`, `GROUPDN`, `ROLEDN`, `LDAPURL`

attrValue Any string that represents an attribute value

Examples of `userattr` Keyword With Various Bind Types

EXAMPLE 6–6 Using the `userattr` Keyword With the `USERDN` Bind Type

The following is an example of the `userattr` keyword associated with a bind based on the user DN:

```
userattr = "manager#USERDN"
```

EXAMPLE 6–6 Using the `userattr` Keyword With the `USERDN` Bind Type *(Continued)*

The bind rule is evaluated to be true if the bind DN matches the value of the `manager` attribute in the targeted entry. You can use this to allow a user's manager to modify employees' attributes. This mechanism only works if the `manager` attribute in the targeted entry is expressed as a full DN.

The following example grants a manager full access to his or her employees' entries:

```
aci: (target="ldap:///dc=example,dc=com")(targetattr="*")
 (version 3.0;acl "manager-write";
 allow (all) userattr = "manager#USERDN";)
```

EXAMPLE 6–7 Using the `userattr` Keyword With the `GROUPDN` Bind Type

The following is an example of the `userattr` keyword associated with a bind based on a group DN:

```
userattr = "owner#GROUPDN"
```

The bind rule is evaluated to be true if the bind DN is a member of the group specified in the `owner` attribute of the targeted entry. For example, you can use this mechanism to allow a group to manage employees' status information. You can use an attribute other than `owner`, as long as the attribute you use contains the DN of a group entry.

The group you point to can be a dynamic group, and the DN of the group can be under any suffix in the directory. However, the evaluation of this type of ACI by the server is very resource intensive.

If you are using static groups that are under the same suffix as the targeted entry, you can use the following expression:

```
userattr = "ldap:///dc=example,dc=com?owner#GROUPDN"
```

In this example, the group entry is under the `dc=example,dc=com` suffix. The server can process this type of syntax more quickly than the previous example.

EXAMPLE 6–8 Using the `userattr` Keyword With the `ROLEDN` Bind Type

The following is an example of the `userattr` keyword associated with a bind based on a role DN:

```
userattr = "exampleEmployeeReportsTo#ROLEDN"
```

EXAMPLE 6–8 Using the `userattr` Keyword With the `ROLEDN` Bind Type *(Continued)*

The bind rule is evaluated to be true if the bind DN belongs to the role specified in the `exampleEmployeeReportsTo` attribute of the targeted entry. For example, if you create a nested role for all managers in your company, you can use this mechanism to grant managers at all levels access to information about employees that are at a lower grade than themselves.

The DN of the role can be under any suffix in the directory. If, in addition, you are using filtered roles, the evaluation of this type of ACI uses a lot of resources on the server.

EXAMPLE 6–9 Using the `userattr` Keyword With the `LDAPURL` Bind Type

The following is an example of the `userattr` keyword associated with a bind based on an LDAP filter:

```
userattr = "myfilter#LDAPURL"
```

The bind rule is evaluated to be true if the bind DN matches the filter specified in the *myfilter* attribute of the targeted entry. The *myfilter* attribute can be replaced by any attribute that contains an LDAP filter.

EXAMPLE 6–10 Using the `userattr` Keyword With Any Attribute Value

The following is an example of the `userattr` keyword associated with a bind based on any attribute value:

```
userattr = "favoriteDrink#Milk"
```

The bind rule is evaluated to be true if the bind DN and the target DN include the `favoriteDrink` attribute with a value of **`Milk`**.

Use of the `userattr` Keyword With the `parent` Keyword for Inheritance

The `userattr` keyword can be used with the `parent` keyword to specify the number of levels below the target that should inherit the ACI. The `userattr` keyword and `parent` keyword use this syntax:

```
userattr = "parent[inheritance_level].attribute#bindType"
```

The `userattr` keyword and `parent` keyword can have the following values:

inheritance_level A comma separated list that indicates how many levels below the target should inherit the ACI. These levels below the targeted entry can be specified: `[0,1,2,3,4]`. Zero (0) indicates the targeted entry.

attribute	The attribute targeted by the `userattr` or `groupattr` keyword.
bindType	The type of bind can be `USERDN` or `GROUPDN`. Inheritance cannot be used with `LDAPURL` and `ROLEDN` binds.

The following example shows how the `userattr` keyword is used with the `parent` keyword for inheritance:

```
userattr = "parent[0,1].manager#USERDN"
```

This bind rule is evaluated to be true if the `bindDN` matches the manager attribute of the targeted entry. The permissions granted when the bind rule is evaluated to be true apply to the target entry *and* to all entries immediately below it.

Use of the `userattr` Keyword to Grant Add Permissions

If you use the `userattr` keyword in conjunction with `all` or `add` permissions, you might find that the behavior of the server is not what you expect. Typically, when a new entry is created in the directory, Directory Server evaluates access rights on the entry being created, and not on the parent entry. However, for ACIs that use the `userattr` keyword, this behavior could create a security hole, and the server's normal behavior is modified to avoid it.

Consider the following example:

```
aci: (target="ldap:///dc=example,dc=com")(targetattr="*")
 (version 3.0; acl "manager-write"; allow (all)
 userattr = "manager#USERDN";)
```

This ACI grants managers all rights on the entries of employees that report to them. However, because access rights are evaluated on the entry being created, this type of ACI would also allow any employee to create an entry in which the manager attribute is set to their own DN. For example, disgruntled employee Joe (`cn=Joe,ou=eng,dc=example,dc=com`), might want to create an entry in the Human Resources branch of the tree, to use (or misuse) the privileges granted to Human Resources employees.

He could do this by creating the following entry:

```
dn: cn= Trojan Horse,ou=Human Resources,dc=example,dc=com
objectclass: top
...
cn: Trojan Horse
manager: cn=Joe,ou=eng,dc=example,dc=com
```

To avoid this type of security threat, the ACI evaluation process does not grant add permission at level 0, that is, to the entry itself. You can, however, use the `parent` keyword to grant add rights below existing entries. You must specify the number of levels below the parent for add

rights. For example, the following ACI allows child entries to be added to any entry in the `dc=example,dc=com` that has a `manager` attribute that matches the bind DN:

```
aci: (target="ldap:///dc=example,dc=com")(targetattr="*")
 (version 3.0; acl "parent-access"; allow (add)
 userattr = "parent[0,1].manager#USERDN";)
```

This ACI ensures that add permission is granted only to users whose bind DN matches the manager attribute of the parent entry.

`ip` Keyword

The `ip` keyword is used to specify that a bind operation must originate from a specific IP address. The `ip` keyword uses this syntax:

`ip = "`*IPaddressList*`" or ip != "`*IPaddressList*`"`

The *IPaddressList* value is a list of one or more comma-separated elements from the following elements:

- A specific IPv4 address: `123.45.6.7`
- An IPv4 address with wildcards to specify a subnetwork: `12.3.45.*`
- An IPv4 address or subnetwork with subnetwork mask: `123.45.6.*, 255.255.255.0`
- An IPv6 address in any of its legal forms and contained in square brackets `[` and `]`, as defined by RFC 2373 (`http://www.ietf.org/rfc/rfc2373.txt`).

  The following addresses are equivalent:

  - `12AB:0000:0000:CD30:0000:0000:0000:0000`
  - `12AB::CD30:0:0:0:0`
  - `12AB:0:0:CD30::`
- An IPv6 address with a subnet prefix length: `12AB::CD30:0:0:0:0/60`

The bind rule is evaluated to be true if the client accessing the directory is located at the named IP address.

The `ip` keyword can be used to force all directory updates to occur from a given machine or network domain. However, the IP address from which a user authenticates can be spoofed, and can therefore not be trusted. Do not base ACIs on this information alone.

The wildcard * can be used to specify a set of IP addresses.

The wildcard * cannot be used in IPv6 addresses.

`dns` Keyword

Note – The `dns` keyword requires that the naming service used on your machine is DNS. If the name service is not DNS, you should use the `ip` keyword instead.

The `dns` keyword is used to specify that a bind operation must originate from a specific domain or host machine. The `dns` keyword uses this syntax:

```
dns = "DNS_Hostname" or dns != "DNS_Hostname"
```

The `dns` keyword requires a fully qualified DNS domain name. Granting access to a host without specifying the domain creates a potential security threat. For example, the following expression is allowed but not recommended:

```
dns = "legend.eng";
```

You should use a fully qualified name such as:

```
dns = "legend.eng.example.com";
```

The `dns` keyword allows wildcards.

```
dns = "*.example.com";
```

The bind rule is evaluated to be true if the client accessing the directory is located in the named domain. This can be useful for allowing access only from a specific domain. Note that wildcards do not work if your system uses a naming service other than DNS.

`timeofday` Keyword

The `timeofday` keyword is used to specify that access can occur at a certain time of day. The time and date on the server are used for the evaluation of the `timeofday` and `dayofweek` bind rules, and not the time on the client. The `timeofday` keyword uses this syntax:

```
timeofday operator "time"
```

The `timeofday` keyword can have the following values:

operator

- Equal to (=)
- Not equal to (!=)
- Greater than or equal to (>=)
- Less than (<)
- Less than or equal to (<=)

time Four digits representing hours and minutes in the 24-hour clock (`0` to `2359`)

- `timeofday = "1200";` is true if the client is accessing the directory during the minute that the system clock shows noon.
- `timeofday != "0100";` is true for access at any other time than 1 a.m.
- `timeofday > "0800";` is true for access from 8:01 a.m. through 11:59 p.m.
- `timeofday >= "0800";` is true for access from 8:00 a.m. through 11:59 p.m.
- `timeofday < "1800";` is true for access from 12:00 midnight through 5:59 p.m.

`dayofweek` Keyword

The `dayofweek` keyword is used to specify that access can occur on a certain day or on certain days of the week. The time and date on the server are used for the evaluation of the `timeofday` and `dayofweek` bind rules, and not the time on the client. The `dayofweek` keyword uses this syntax:

`dayofweek = "`*day1, day2 ...*`"`

The bind rule is true if the directory is being accessed on one of the days listed.

The `dayofweek` keyword can have one or more of the following values: `sun`, `mon`, `tue`, `wed`, `thu`, `fri`, `sat`.

`authmethod` Keyword

The `authmethod` keyword is used to specify that a client must bind to the directory by using a specific authentication method. The `authmethod` keyword uses this syntax:

`authmethod = "`*authentication_method*`"`

The `authmethod` keyword can have the following values:

None	Authentication is not checked during bind rule evaluation. This is the default.
Simple	The bind rule is evaluated to be true if the client is accessing the directory using a username and password.
SSL	The client must bind to the directory over a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) connection. The bind rule is evaluated to be true if the client authenticates to the directory by using a certificate over LDAPS. It will not be true if the client authenticates by using simple authentication (bind DN and password) over LDAPS.

SASL *sasl_mechanism*	The bind rule is evaluated to be true if the client authenticates to the directory by using one of the following SASL mechanisms: `DIGEST-MD5`, `GSSAPI`, or `EXTERNAL`.

Boolean Bind Rules

Bind rules can be complex expressions that use the Boolean expressions `AND`, `OR`, and `NOT` to set precise access rules. Boolean bind rules use this syntax:

`(`*bindRuleA* `and (`*bindRuleB* `or (`*bindRuleC* `and` *bindRuleD*`));)`

Parentheses defines the order in which rules are evaluated, and a trailing semicolon must appear after the final rule.

EXAMPLE 6–11 Boolean Bind Rule

The bind rule is true if both of the following conditions are met:

- The bind DN client is accessed from within the `example.com` domain
- The bind DN client is a member of either the administrators group *or* the bind DN client a member of both the mail administrators and calendar administrators groups

```
(dns = "*.example.com"
  and (groupdn = "ldap:///cn=administrators, dc=example,dc=com"
  or (groupdn = "ldap:///cn=mail administrators, dc=example,dc=com"
  and groupdn = "ldap:///cn=calendar administrators, dc=example,dc=com"));)
```

▼ To Allow Normal Users to Manage User Accounts Using `dsutil` Command

- **Add the following ACIs to allow users to manage other user accounts using the `dsutil` command.**

```
$ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w - -c
dn: cn=config
changetype: modify
add: aci
aci: (targetattr="*")(version 3.0; acl "Allow the Suffix Manager to browse the tree"; \
allow (read,search,compare)userdn = "ldap:///$USERSFXADMIN";)
aci: (targetattr="nsslapd-rootpw")\
(version 3.0; acl "Prevent the Suffix Manager from accessing passwords"; \
deny (all)userdn = "ldap:///$USERSFXADMIN";)
aci: (targetattr="userPassword")\
(version 3.0; acl "Prevent the Suffix Manager from accessing passwords"; \
```

```
deny (all)userdn = "ldap:///$USERSFXADMIN";)
aci: (targetattr="dsKeyedPassword")\
(version 3.0; acl "Prevent the Suffix Manager from accessing passwords"; \
deny (all)userdn = "ldap:///$USERSFXADMIN";)
```

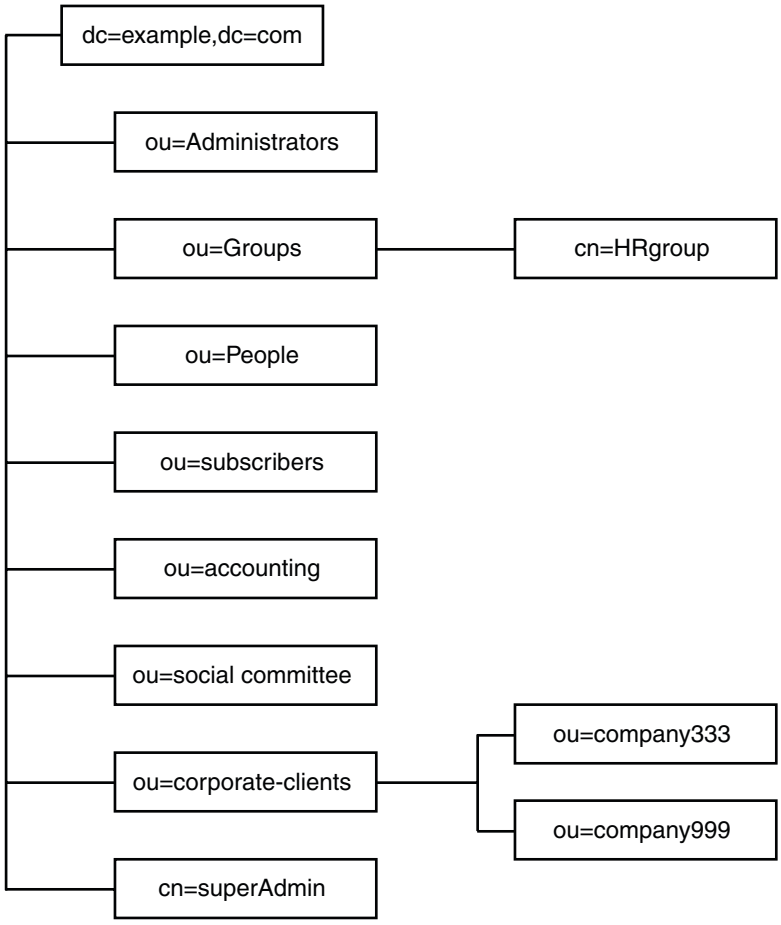
For more information about `dsutil` command, see `dsutil(1M)`.

Access Control Usage Examples

The examples in this section illustrate how an imaginary ISP company, Example.com, would implement its access control policy.

In addition, you can find ACI examples in the example LDIF file provided with your installation, *install_path*`/resources/ldif/Example.ldif`.

All of the examples explain how to perform a given task by using an LDIF file. The following figure shows the example.com Directory Information Tree in graphical form.



Example.com offers a web hosting service and internet access. Part of Example.com's web-hosting service is to host the directories of client companies. Example.com actually hosts and partially manages the directories of two medium-sized companies, Company333 and Company999. Example.com also provides internet access to a number of individual subscribers.

Example.com wants to put the following access rules in place:

- Grant anonymous read, search, and compare access to the entire Example.com tree for Example.com employees. See "Granting Anonymous Access" on page 165.
- Grant write access to Example.com employees for personal information, such as `homeTelephoneNumber`, and `homeAddress`. See "Granting Write Access to Personal Entries" on page 166.

- Grant Example.com subscribers the right to read the entry `dc=example,dc=com` for company contact information, but not to read any entries below it. See "Granting Access to a Certain Level" on page 167.
- Grant Example.com employees the right to add any role to their entry, except certain critical roles. See "Restricting Access to Key Roles" on page 168.
- Grant certain administrators the same rights as the Directory Manager for a suffix. See "Granting a Role Full Access to an Entire Suffix" on page 169.
- Grant the Example.com Human Resources group all rights on the entries in the People branch. See "Granting a Group Full Access to a Suffix" on page 169.
- Grant all Example.com employees the right to create group entries under the Social Committee branch of the directory, and to delete group entries that an employee owns. See "Granting Rights to Add and Delete Group Entries" on page 170.
- Grant all Example.com employees the right to add themselves to group entries under the Social Committee branch of the directory. See "Allowing Users to Add or Remove Themselves From a Group" on page 171.
- Grant access to the directory administrator (role) of Company333 and Company999 on their respective branches of the directory tree, with certain conditions. These conditions include SSL authentication, time and date restrictions, and specified location. See "Granting Conditional Access to a Group or Role" on page 172.
- Grant individual subscribers access to their own entries. See "Granting Write Access to Personal Entries" on page 166.
- Deny individual subscribers access to the billing information in their own entries. See "Denying Access" on page 173.
- Grant anonymous access to the world to the individual subscribers subtree, except for subscribers who have specifically requested to be unlisted. If desired, this part of the directory could be a read-only server outside of the firewall, and be updated once a day. See "Granting Anonymous Access" on page 165 and "Setting a Target Using Filtering" on page 175.

Granting Anonymous Access

Most directories are configured to enable you to anonymously access at least one suffix for read, search, or compare. You might want to set these permissions if you are running a corporate personnel directory, such as a phone book that you want employees to be able to search. This is the case at Example.com internally, as shown in "ACI "Anonymous Example.com"" on page 166.

As an ISP, Example.com also wants to advertise the contact information of all of its subscribers by creating a public phone book that is accessible to the world. This is depicted in "ACI "Anonymous World"" on page 166.

ACI "Anonymous Example.com"

In LDIF, to grant read, search, and compare permissions to the entire Example.com tree to Example.com employees, you would write the following statement:

```
aci: (targetattr !="userPassword")(version 3.0; acl "Anonymous
 example"; allow (read, search, compare)
 userdn= "ldap:///anyone") ;)
```

This example assumes that the `aci` is added to the `dc=example,dc=com` `entry`. Note that the `userPassword` attribute is excluded from the scope of the ACI.

Note – Protect attributes that are confidential and attributes that should not be visible using the same syntax used in the example to protect the password attribute, `(targetattr !="`*attribute-name*`")`.

ACI "Anonymous World"

In LDIF, to grant read and search access of the individual subscribers subtree to the world, while denying access to information on subscribers who want to be unlisted, you could write the following statement:

```
aci: (targetfilter= "(!(unlistedSubscriber=yes))")
 (targetattr="homePostalAddress || homePhone || mail")
 (version 3.0; acl "Anonymous World"; allow (read, search)
 userdn="ldap:///anyone";)
```

This example assumes that the ACI is added to the `ou=subscribers,dc=example, dc=com` entry. The example also assumes that every subscriber entry has an `unlistedSubscriber` attribute that is set to `yes` or `no`. The target definition filters out the unlisted subscribers based on the value of this attribute. For details on the filter definition, refer to "Setting a Target Using Filtering" on page 175.

Granting Write Access to Personal Entries

Many directory administrators want to allow internal users to change some but not all of the attributes in their own entry. The directory administrators at Example.com want to allow users to change their own password, home telephone number, and home address, but nothing else. This is depicted in "ACI "Write Example.com"" on page 167.

Example.com also has a policy to let their subscribers update their own personal information in the Example.com tree provided that the subscribers establish an SSL connection to the directory. This is depicted in "ACI "Write Subscribers"" on page 167.

ACI "Write Example.com"

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

In LDIF, to grant Example.com employees the right to update their home telephone number and home address, you would write the following statement:

```
aci: (targetattr="homePhone ||
 homePostalAddress")(version 3.0; acl "Write Example.com";
 allow (write) userdn="ldap:///self" ;)
```

This example assumes that the ACI is added to the `ou=People,dc=example,dc=com` entry.

ACI "Write Subscribers"

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

In LDIF, to grant Example.com subscribers the right to update their home telephone number, you would write the following statement:

```
aci: (targetattr="homePhone")
 (version 3.0; acl "Write Subscribers"; allow (write)
 userdn= "ldap://self" and authmethod="ssl";)
```

This example assumes that the `aci` is added to the `ou=subscribers,dc=example, dc=com` entry, and that users must bind using SSL.

Note that Example.com subscribers do not have write access to their home address because they might delete that attribute. The home address is business-critical information that Example.com needs for billing purposes.

Granting Access to a Certain Level

You can set the scope of an ACI to affect different levels within your directory tree, to fine-tune the level of access you want to allow. The target ACI scope can be set to one of the following:

`base`	The entry itself
`onelevel`	The entry itself and all entries one level below
`subtree`	The entry itself and all entries beneath that entry, to an unlimited depth

ACI "Read Example.com only"

In LDIF, to grant Example.com subscribers the right to read the entry `dc=example,dc=com` for company contact information, but not allow access to any entries below it, you would write the following statement:

```
aci: (targetscope="base") (targetattr="*")(version 3.0;
 acl "Read Example.com only";  allow (read,search,compare)
 userdn="ldap:///cn=*,ou=subscribers,dc=example,dc=com";)
```

This example assumes that the ACI is added to the `dc=example, dc=com` entry.

Restricting Access to Key Roles

You can use role definitions in the directory to identify functions that are critical to your business, such as the administration of your network and directory.

For example, you might create a `superAdmin` role by identifying a subset of your system administrators who are available at a particular time of day and day of the week at corporate sites worldwide. Or you might want to create a `First Aid` role that includes all staff members who have first aid training at a particular site. For information about creating role definitions see "Managing Roles" on page 236.

When a role gives any sort of privileged user rights over critical corporate or business functions, consider restricting access to that role. For example, at Example.com, employees can add any role to their own entry, except the `superAdmin` role, as shown in the following example.

ACI "Roles"

In LDIF, to grant Example.com employees the right to add any role to their own entry, except the `superAdmin` role, you would write the following statement:

```
aci: (targetattr="*") (targattrfilters="add=nsRoleDN:
 (nsRoleDN !="cn=superAdmin, dc=example, dc=com")")
 (version 3.0; acl "Roles"; allow (write)
 userdn= "ldap:///self" ;)
```

This example assumes that the ACI is added to the `ou=People,dc=example, dc=com` entry.

Granting a Role Full Access to an Entire Suffix

Sometimes it is useful to grant certain users the same rights as the Directory Manager for a suffix. At Example.com, Kirsten Vaughan is an administrator for Directory Server. She has a role of `superAdmin`. This role has the following advantages:

- Better security because administrators binding as themselves can be forced to use strong authentication such as SSL
- Better security because the Directory Manager password is known by fewer people
- More traceability through logging

Note – Adding Kirsten Vaughan to the `cn=Administrators,cn=config` group would also grant her the same rights as Directory Manager.

To give a user the same rights as the Directory Manager for the whole server, follow the procedure in "To Create an Administration User with Root Access" on page 62.

ACI "Full Access"

In LDIF, to grant the administrator Kirsten Vaughan the same rights as a Directory Manager, use the following statement:

```
aci: (targetattr="*") (version 3.0; acl "Full Access";
 allow (all) groupdn= "ldap:///cn=SuperAdmin,dc=example,dc=com"
 and authmethod="ssl" ;)
```

This example assumes that the ACI is added to the root entry "" (no text).

Granting a Group Full Access to a Suffix

Most directories have groups that are used to identify certain corporate functions. A group can be given access to all or part of the directory. By applying access rights to a group, you can avoid setting access rights for each member individually. Instead, you grant users access rights by adding them to a group.

For example, when you create a Directory Server instance, an Administrators group `cn=Administrators,cn=config` with full access to the directory is created by default.

At Example.com, the Human Resources group is allowed full access to the `ou=People` branch of the directory so that they can update the employee directory, as shown in "ACI "HR"" on page 170.

ACI "HR"

In LDIF, to grant the HR group all rights to the employee branch of the directory, you would use the following statement:

```
aci: (targetattr="*") (version 3.0; acl "HR"; allow (all)
  groupdn= "ldap:///cn=HRgroup,ou=Groups,dc=example,dc=com";)
```

This example assumes that the ACI is added to the following entry:

```
ou=People,dc=example,dc=com
```

Granting Rights to Add and Delete Group Entries

Some organizations allow employees to create entries in the tree to increase employees' efficiency and to encourage employees to contribute to the corporate dynamics. At Example.com, for example, the social committee is organized into various clubs, such as tennis, swimming, skiing, and role-playing.

Any Example.com employee can create a group entry that represents a new club, as shown in "ACI "Create Group"" on page 170.

Any Example.com employee can become a member of one of these groups, as shown in "Allowing Users to Add or Remove Themselves From a Group" on page 171.

Only the group owner can modify or delete a group entry, as shown in "ACI "Delete Group"" on page 171.

ACI "Create Group"

In LDIF, to grant Example.com employees the right to create a group entry under the `ou=Social Committee` branch, you would write the following statement:

```
aci: (targetattr="*") (targattrfilters="add=objectClass:
(|(objectClass=groupOfNames)(objectClass=top))")
(version 3.0; acl "Create Group"; allow (read,search,add)
userdn= "ldap:///uid=*,ou=People,dc=example,dc=com")
and dns="*.Example.com";)
```

This example assumes that the ACI is added to the `ou=Social Committee,dc=example,dc=com` entry.

Note –

- This ACI does not grant write permission, which means that the entry creator cannot modify the entry.
- Because the server adds the value `top` behind the scenes, you need to specify `objectClass=top` in the `targattrfilters` keyword.
- This ACI restricts the client machine to be in the `example.com`domain.

ACI "Delete Group"

In LDIF, to grant Example.com employees the right to modify or delete a group entry of the group to which they belong under the `ou=Social Committee` branch, you would write the following statement:

```
aci: (targetattr = "*") (targattrfilters="del=objectClass:
(objectClass=groupOfNames)")
 (version 3.0; acl "Delete Group"; allow (write,delete)
 userattr="owner#GROUPDN";)
```

This example assumes that the `aci` is added to the `ou=Social Committee,dc=example,dc=com` entry.

Note that to use DSCC to create this ACI is not very effective because you have to use manual editing mode to create the target filter and to check group ownership.

Allowing Users to Add or Remove Themselves From a Group

Many directories set ACIs that allow users to add or remove themselves from groups such as mailing lists.

At Example.com, employees can add themselves to any group entry under the `ou=Social Committee` subtree, as shown in "ACI "Group Members"" on page 171.

ACI "Group Members"

In LDIF, to grant Example.com employees the right to add themselves to a group, you would write the following statement:

```
aci: (targettattr="member")(version 3.0; acl "Group Members";
 allow (selfwrite)
 (userdn= "ldap:///uid=*,ou=People,dc=example,dc=com") ;)
```

This example assumes that the ACI is added to the `ou=Social Committee, dc=example,dc=com` entry.

Granting Conditional Access to a Group or Role

In many cases, when you grant a group or role privileged access to the directory, you must ensure that those privileges are protected from intruders trying to impersonate your privileged users. Therefore, in many cases, access control rules that grant critical access to a group or role are often associated with a number of conditions.

Example.com, for example, has created a Directory Administrator role for each of its hosted companies, Company333 and Company999. Example.com wants these companies to be able to manage their own data and implement their own access control rules while securing the data against intruders.

For this reason, Company333 and Company999 have full rights on their respective branches of the directory tree, provided that the following conditions are fulfilled:

- The connection is authenticated using a certificate over SSL.
- Access is requested between 8:00 and 18:00, Monday through Thursday.
- Access is requested from a specified IP address for each company.

These conditions are depicted in one ACI for each company, ACI “Company333” and ACI “Company999”. Because the content of both ACIs is the same, the following examples use the “Company333” ACI only.

ACI “Company333”

In LDIF, to grant Company333 full access to its own branch of the directory under the conditions stated previously, you would write the following statement:

```
aci: (targetattr = "*") (version 3.0; acl "Company333"; allow (all)
 (roledn="ldap:///cn=DirectoryAdmin,ou=Company333,
 ou=corporate clients,dc=example,dc=com") and (authmethod="ssl")
 and (dayofweek="Mon,Tues,Wed,Thu") and (timeofday >= "0800" and
 timeofday <= "1800") and (ip="255.255.123.234"); )
```

This example assumes that the ACI is added to the `ou=Company333,ou=corporate clients,dc=example,dc=com` entry.

Denying Access

If you have already allowed access to a large part of your suffix, you might want to deny access to a smaller part of the suffix beneath the existing ACI.

Note – Denying access should be avoided where possible, because it can lead to surprising or complicated access control behavior. Restrict access by using a combination of scoping, attribute lists, target filters and so on.

Also, deleting a deny access ACI does not remove rights, but instead expands the rights set by other ACIs.

When the Directory Server evaluates access rights, it reads `deny` rights first, then `allow` rights.

In the examples that follow, Example.com wants all subscribers to be able to read billing information, such as connection time or account balance, under their own entries. Example.com also explicitly wants to deny write access to that information. The read access is depicted in "ACI "Billing Info Read"" on page 173. The `deny` access is depicted in "ACI "Billing Info Deny"" on page 173.

ACI "Billing Info Read"

In LDIF, to grant subscribers permission to read billing information in their own entry, you would write the following statement:

```
aci: (targetattr="connectionTime || accountBalance")
 (version 3.0; acl "Billing Info Read"; allow (search,read)
  userdn="ldap:///self";)
```

This example assumes that the relevant attributes have been created in the schema and that the ACI is added to the `ou=subscribers,dc=example,dc=com` entry.

ACI "Billing Info Deny"

In LDIF, to deny subscribers permission to modify billing information in their own entry, you would write the following statement:

```
aci: (targetattr="connectionTime || accountBalance")
 (version 3.0; acl "Billing Info Deny";
 deny (write) userdn="ldap:///self";)
```

This example assumes that the relevant attributes have been created in the schema and that the ACI is added to the `ou=subscribers,dc=example,dc=com` entry.

Proxy Authorization

The proxy authorization method is a special form of authentication. A user that binds to the directory by using his or her own identity is granted the rights of another user through proxy authorization.

To configure Directory Server to allow proxy requests you must do the following:

- Grant the administrators the right to proxy as other users.
- Grant your regular users normal access rights as defined in your access control policy.

Note – You can grant proxy rights to any users of the directory except the Directory Manager. In addition, you cannot use the Directory Manager's DN as a proxy DN. You need to exercise great care when granting proxy rights because you grant the right to specify any DN (except the Directory Manager DN) as the proxy DN. If Directory Server receives more than one proxied authentication control in the same operation, an error is returned to the client application and the operation attempt is unsuccessful.

Example Proxy Authorization

Example.com wants the client application that binds as `MoneyWizAcctSoftware` to have the same access rights to the LDAP data as an Accounting Administrator.

The following parameters apply:

- The client application's bind DN is `uid=MoneyWizAcctSoftware, ou=Applications,dc=example,dc=com`.
- The targeted subtree to which the client application is requesting access is `ou=Accounting,dc=example,dc=com`.
- An Accounting Administrator with access permissions to the `ou=Accounting,dc=example,dc=com` subtree exists in the directory.

For the client application to gain access to the Accounting subtree, by using the same access permissions as the Accounting Administrator, the following must be true:

- The Accounting Administrator must have access permissions to the `ou=Accounting,dc=example,dc=com` subtree. For example, the following ACI grants all rights to the Accounting Administrator entry:

```
aci: (targetattr="*") (version 3.0; acl "allowAll-AcctAdmin"; allow
 (all) userdn="ldap:///uid=AcctAdministrator,ou=Administrators,
 dc=example,dc=com";)
```

- The following ACI that grants proxy rights to the client application must exist in the directory:

```
aci: (targetattr="*") (version 3.0; acl "allowproxy- accountingsoftware";
 allow (proxy) userdn= "ldap:///uid=MoneyWizAcctSoftware,ou=Applications,
dc=example,dc=com";)
```

With this ACI in place, the `MoneyWizAcctSoftware` client application can bind to the directory and then send an LDAP command, such as `ldapsearch` or `ldapmodify`, that requires the access rights of the proxy DN.

In this example, if the client wanted to perform an `ldapsearch` command, the command would include the following controls:

```
$ ldapsearch -D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" -w - \
 -Y "dn: uid=AcctAdministrator,ou=Administrators,dc=example,dc=com" ...
```

If the client wanted to perform an `ldapmodify` command, the command would include the following controls:

```
$ ldapmodify -h hostname -p port \
-D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" -w - \
-Y"dn: uid=AcctAdministrator,ou=Administrators,dc=example,dc=com"
dn: uid=AcctAdministrator,ou=Administrators,dc=example,dc=com
changetype: modify
delete: userpassword
-
add: userpassword
userpassword: admin1
```

Note that the client binds as itself, but is granted the privileges of the proxy entry. The client does not need the password of the proxy entry.

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 might want to use a filter to set the target.

In LDIF, to use a filter to allow all users in HR access to employee entries, you would write the following statement:

```
aci: (targetattr="*") (targetfilter=(objectClass=employee))
 (version 3.0; acl "HR access to employees";
 allow (all) groupdn= "ldap:///cn=HRgroup,ou=People,dc=example,dc=com";)
```

This example assumes that the ACI is added to the `ou=People,dc=example,dc=com` entry.

Note – Because search filters do not directly name the object for which you are managing access, try not to allow or deny access to the wrong objects. Unintentionally allowing or denying access to the wrong objects becomes more of a risk as your directory becomes more complex. Additionally, filters can make it difficult for you to troubleshoot access control problems within your directory.

Defining Permissions for DNs That Contain a Comma

DNs that contain commas require special treatment 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=Example.com Bolivia\, S.A.
objectClass: top
objectClass: organization
aci: (target="ldap:///o=Example.com Bolivia\,S.A.") (targetattr="*")
(version 3.0; acl "aci 2"; allow (all) groupdn =
"ldap:///cn=Directory Administrators, o=Example.com Bolivia\, S.A.";)
```

Viewing Effective Rights

When maintaining the access policy on the entries of a directory, you need to know the effects on security of the ACIs that you define. Directory Server enables you to evaluate existing ACIs by viewing the effective rights that the ACIs grant for a given user on a given entry.

Directory Server responds to the "Get Effective Rights", control which can be included in a search operation. The response to this control is to return the effective rights information about the entries and attributes in the search results. This extra information includes read and write permissions for each entry and for each attribute in each entry. The permissions can be requested for the bind DN that is used for the search or for an arbitrary DN. This choice allows administrators to test the permissions of directory users.

Effective rights functionality relies on an LDAP control. You must ensure that the proxy identity used to bind to the remote server is also allowed to access the effective rights attributes.

Restricting Access to the Get Effective Rights Control

The operation of viewing effective rights is a directory operation that needs to be protected and appropriately restricted.

To restrict access to effective rights information, modify the default ACI for `getEffectiveRights` attribute. Then create a new ACI for the `getEffectiveRightsInfo` attribute .

For example, the following ACI allows only members of the Directory Administrators Group to get effective rights:

```
aci: (targetattr != "aci")(version 3.0; acl
 "getEffectiveRights"; allow(all) groupdn =
 "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)
```

To obtain effective rights information, you need to have access control rights to use the Effective Rights control *and* read access to the `aclRights` attribute. This double layer of access control provides basic security that can be more finely tuned if necessary. By analogy with proxy, if you have read access to the `aclRights` attribute in an entry, you can request information about anyone's rights to that entry and its attributes. This implies that the user who manages the resource can determine who has rights to that resource, even if that user does not actually manage those with the rights.

If a user requesting rights information does not have the rights to use the Effective Rights control, the operation fails and an error message is returned. However, if the user requesting rights information does have the rights to use the control but lacks the rights to read the `aclRights` attribute, the `aclRights` attribute will not appear in the returned entry. This behavior reflects Directory Server's general search behavior.

Using the Get Effective Rights Control

Specify the "Get Effective Rights" control by using the `ldapsearch` command with the `-J "1.3.6.1.4.1.42.2.27.9.5.2"` option. By default, the control returns the effective rights of the bind DN entry on the entries and attributes in the search results.

Use the following options to change the default behavior:

- `-c "dn:` *bind DN* `"` — The search results show the effective rights of the user binding with the given *DN*. This option allows an administrator to check the effective rights of another user. The option `-c "dn:"` shows the effective rights for anonymous authentication.
- `-X "`*attributeName ...* `"` — The search results also include the effective rights on the named attributes. Use this option to specify attributes that do not appear in the search results. For example, use this option to determine if a user has permission to add an attribute that does not currently exist in an entry.

- When using either or both of the `-c` and `-X` options, the `-J` option with the OID of the "Get Effective Rights" control is implied and does not need to be specified. If you specify a NULL value for the Effective Rights control, the rights are retrieved for the current user. In addition, the rights are retrieved for the attributes and entries that are being returned with the current `ldapsearch` operation.

  Then you must select the type of information you want to view. Choose either the simple rights or the more detailed logging information that explains how those rights are granted or denied. The type of information is determined by adding either `aclRights` or `aclRightsInfo`, respectively, as an attribute to return in the search results. You can request both attributes to receive all effective rights information, although the simple rights essentially repeat the information in the detailed logging information.

Note – The `aclRights` and `aclRightsInfo` attributes behave like virtual operational attributes. These attributes are not stored in the directory and are not returned unless explicitly requested. They are generated by Directory Server in response to the "Get Effective Rights" control.

Thus, these attributes cannot be used in filters or search operations of any kind.

The effective rights feature inherits other parameters that affect access control. These parameters include time of day, authentication method, machine address, and name.

The following example shows how the user Carla Fuente can view her rights in the directory. In the results, `1` means that permission is granted, and `0` means that permission is denied.

```
$ ldapsearch -J "1.3.6.1.4.1.42.2.27.9.5.2" -h host1.Example.com -p 389 \
 -D "uid=cfuente,ou=People,dc=example,dc=com" -w - -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights

Enter bind password:
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

This result shows Carla Fuente the entries in the directory where she has at least read permission and shows that she can modify her own entry. The Effective Rights control does not bypass normal access permissions, so a user does not see the entries for which they do not have read permission. In the following example, the Directory Manager can see the entries to which Carla Fuente does not have read permission:

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights

Enter bind password:
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Directory Administrators, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
dn: ou=Special Users,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

In the preceding output, the Directory Manager can see that Carla Fuente cannot even view the Special Users or the Directory Administrators branches of the directory tree. In the following example, the Directory Manager can see that Carla Fuente cannot modify the `mail` and `manager` attributes in her own entry:

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(uid=cfuente)" aclRights "*"

Enter bind password:
version: 1
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;attributeLevel;mail: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
mail: cfuente@Example.com
aclRights;attributeLevel;uid: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
```

```
uid: cfuente
aclRights;attributeLevel;givenName: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
givenName: Carla
aclRights;attributeLevel;sn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
sn: Fuente
aclRights;attributeLevel;cn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
cn: Carla Fuente
aclRights;attributeLevel;userPassword: search:0,read:0,
 compare:0,write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
userPassword: {SSHA}wnbWHIq2HPiY/5ECwe6MWBGx2KMiZ8JmjF80Ow==
aclRights;attributeLevel;manager: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
manager: uid=bjensen,ou=People,dc=example,dc=com
aclRights;attributeLevel;telephoneNumber: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
telephoneNumber: (234) 555-7898
aclRights;attributeLevel;objectClass: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

Advanced Access Control: Using Macro ACIs

Organizations that use repeating directory tree structures can optimize the number of ACIs used in the directory by using macros. When you reduce the number of ACIs in your directory tree, it is easier to manage your access control policy. In addition, the efficiency of your ACI memory usage is improved.

Macros are placeholders that are used to represent a DN or a portion of a DN in an ACI. You can use a macro to represent a DN in the target portion of the ACI, in the bind rule portion, or in both. In practice, when Directory Server gets an incoming LDAP operation, the ACI macros are matched against the resource targeted by the LDAP operation. The matching occurs in order to determine a matching substring, if any matching substring exists. If a match exists, the bind rule-side macro is expanded using the matched substring, and access to the resource is determined by evaluating that expanded bind rule.

This section contains an example of a macro ACI and information about macro ACI syntax.

Macro ACI Example

The benefits of macro ACIs and how they work are best explained by using an example. Figure 6–1 shows a directory tree in which using macro ACIs is an effective way of reducing the overall number of ACIs.

In this illustration, note the repeating pattern of subdomains with the same tree structure (`ou=groups,ou=people`). This pattern is also repeated across the tree because the Example.com directory tree stores the two suffixes `dc=hostedCompany2,dc=example,dc=com`, and `dc=hostedCompany3,dc=example,dc=com`, which are not shown in the figure.

The ACIs in the directory tree also have a repeating pattern. For example, the following ACI is located on the `dc=hostedCompany1,dc=example,dc=com` node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))(version 3.0;
 acl "Domain access"; allow (read,search) groupdn=
 "ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,
 dc=example,dc=com";)
```

This ACI grants the `domainAdmins` group read and search rights to any entry in the `dc=hostedCompany1,dc=example,dc=com` tree.

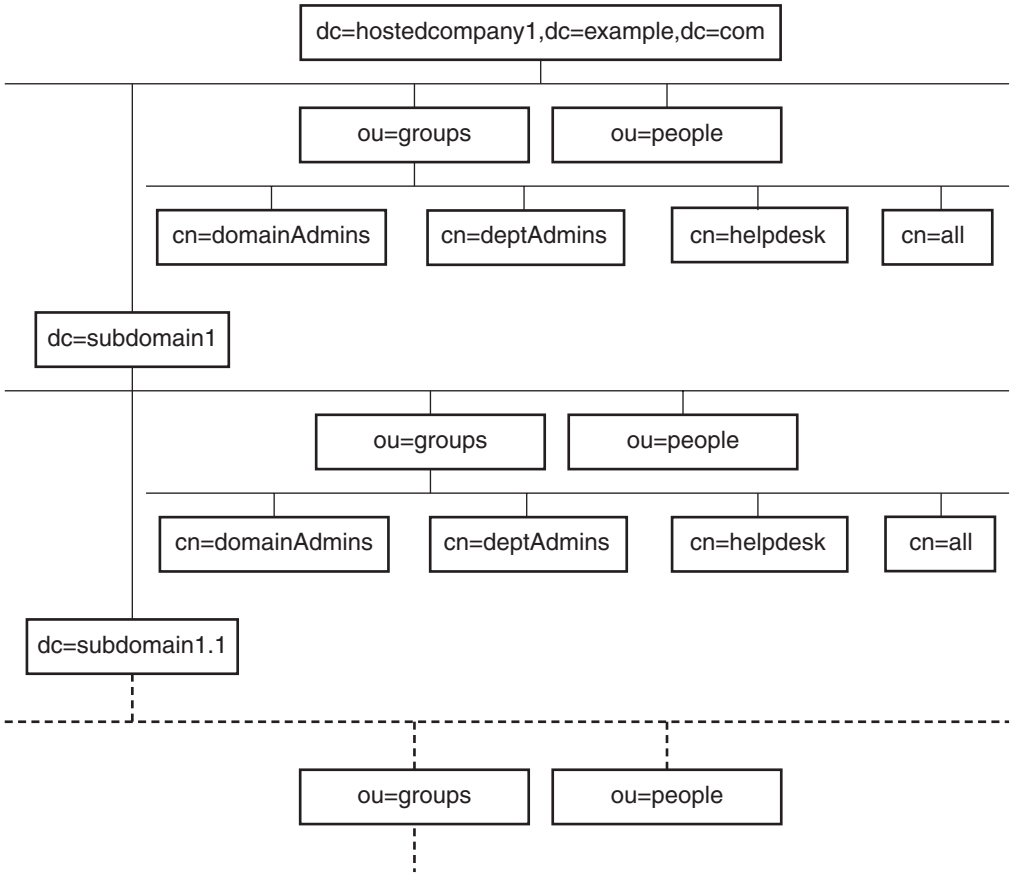


FIGURE 6–1 Example Directory Tree for Macro ACIs

The following ACI is located on the `dc=hostedCompany1,dc=example,dc=com` node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com";)
```

The following ACI is located on the `dc=subdomain1,dc=hostedCompany1, dc=example,dc=com` node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany1,
  dc=example,dc=com";)
```

The following ACI is located on the `dc=hostedCompany2,dc=example,dc=com` node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany2, dc=example,dc=com";)
```

The following ACI is located on the `dc=subdomain1,dc=hostedCompany2, dc=example,dc=com` node:

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany2,
 dc=example,dc=com";)
```

In the preceding four ACIs, the only difference is the DN that is specified in the `groupdn` keyword. By using a macro for the DN, it is possible to replace these ACIs with a single ACI at the root of the tree, on the `dc=example,dc=com` node. This macro ACI reads as follows:

```
aci: (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

Note that the `target` keyword which was not previously used needs to be introduced.

In the preceding example, the number of ACIs is reduced from four to one. However, the real benefit is a factor of how many repeating patterns you have down and across your directory tree.

Macro ACI Syntax

To simplify the discussion in this section, the ACI keywords such as `userdn`, `roledn`, `groupdn`, and `userattr` that are used to provide bind credentials are collectively called the *subject* of the ACI. The subject determines to whom the ACI applies.

The following table shows which macros can be used to replace specific ACI keywords.

TABLE 6–3 Macro ACI Keywords

Macro	Description	ACI Keywords
`($dn)`	For matching in the target, and direct substitution in the subject.	`target, targetfilter, userdn, roledn, groupdn, userattr`

TABLE 6–3 Macro ACI Keywords *(Continued)*

Macro	Description	ACI Keywords
`[$dn]`	For substituting multiple RDNs that work in subtrees of the subject.	`targetfilter, userdn, roledn, groupdn, userattr`
`($attr.`*attrName*`)`	For substituting the value of the *attributeName* attribute from the target entry into the subject.	`userdn, roledn, groupdn, userattr`

The following restrictions apply to macro ACI keywords:

- When using the `($dn)` and `[$dn]` macros in a subject, you *must* define a target that contains the `($dn)` macro.
- You can combine the `($dn)` macro (but not the `[$dn]` macro) with the `($attr.`*attrName*`)` macro in a subject.

Matching for `($dn)` in the Target

The `($dn)` macro in the target of an ACI determines the substitution value by comparing it to the entry targeted by the LDAP request. For example, you have an LDAP request targeted at this entry:

```
cn=all,ou=groups,dc=subdomain1, dc=hostedCompany1,dc=example,dc=com
```

In addition, you have an ACI that defines the target as follows:

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

The `($dn)` macro matches with "`dc=subdomain1, dc=hostedCompany1`". This substring is then used for substitutions in the subject of the ACI.

Substituting `($dn)` in the Subject

In the subject of the ACI, the `($dn)` macro is replaced by the entire substring that matches in the target. For example:

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),dc=example,dc=com"
```

The subject becomes this:

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,
 dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"
```

After the macro has been expanded, Directory Server evaluates the ACI following the normal process to determine whether access is granted.

Note – Unlike a standard ACI, an ACI that uses macro substitution does not necessarily grant access to the child of the targeted entry. This is because when the child DN is the target, the substitution might not create a valid DN in the subject string.

Substituting `[$dn]` in the Subject

The substitution mechanism for `[$dn]` is slightly different than for `($dn)`. The DN of the targeted resource is examined several times, each time dropping the left-most RDN component, until a match is found.

For example, suppose that you have an LDAP request targeted at the `cn=all,ou=groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com` subtree, and the following ACI:

```
aci: (targetattr="*")
 (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],
 dc=example,dc=com";)
```

The server proceeds as follows to expand this ACI:

1. The server verifies that the `($dn)` in target matches `dc=subdomain1,dc=hostedCompany1`.
2. The server replaces `[$dn]` in the subject with `dc=subdomain1,dc=hostedCompany1`.

   The resulting subject is `groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"`. If access is granted because the bind DN is a member of that group, the macro expansion stops, and the ACI is evaluated. If the bind DN is not a member, the process continues.
3. The server replaces `[$dn]` in the subject with `dc=hostedCompany1`.

   The resulting subject is `groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com"`. Again, the bind DN is tested as a member of this group and if it is, the ACI is evaluated fully. If the bind DN is not a member, macro expansion stops with the last RDN of the matched value, and ACI evaluation is finished for this ACI.

The advantage of the `[$dn]` macro is that it provides a flexible way to grant domain-level administrators access to *all* the subdomains in the directory tree. Therefore, the `[$dn]` macro is useful for expressing a hierarchical relationship between domains.

For example, consider the following ACI:

```
aci: (target="ldap:///ou=*,($dn),dc=example,dc=com") (targetattr="*")
(targetfilter=(objectClass=nsManagedDomain))
(version 3.0; acl "Domain access"; allow (read,search) groupdn=
"ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

The ACI grants access to the members of `cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com` to all of the subdomains under `dc=hostedCompany1`. Thus, an administrator who belongs to that group could access, for example, the subtree `ou=people,dc=subdomain1.1,dc=subdomain1`.

However, at the same time, members of `cn=DomainAdmins,ou=Groups, dc=subdomain1.1` would be denied access to the `ou=people,dc=subdomain1, dc=hostedCompany1` and `ou=people,dc=hostedCompany1` nodes.

Macro Matching for (`$attr.`*attrName*`)`

The (`$attr.`*attrname*`)` macro is always used in the subject part of a DN. For example, you could define the following `roledn`:

```
roledn = "ldap:///cn=DomainAdmins,($attr.ou),dc=HostedCompany1,dc=example,dc=com"
```

Now, assume that the server receives an LDAP operation that is targeted at the following entry:

```
dn: cn=Babs Jensen,ou=People,dc=HostedCompany1,dc=example,dc=com
cn: Babs Jensen
sn: Jensen
ou: Sales
...
```

To evaluate the `roledn` part of the ACI, the server reads the value of the `ou` attribute stored in the targeted entry. The server then substitutes this value in the subject to expand the macro. In the example, the `roledn` is expanded as follows:

```
roledn = "ldap:///cn=DomainAdmins,ou=Sales,dc=HostedCompany1,dc=example,dc=com"
```

Directory Server then evaluates the ACI according to the normal ACI evaluation algorithm.

When the attribute that is named in the macro is multivalued, each value is used in turn to expand the macro. The first value that provides a successful match is used.

Logging Access Control Information

To obtain information about access control in the error logs, you must set the appropriate log level.

▼ To Set Logging for ACIs

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Set the log level to take into account ACI processing.**

  ```
  $ dsconf set-log-prop -h host -p port error level:err-acl
  ```

Client-Host Access Control Through TCP Wrapping

You can control the host or IP address from which connections are accepted or rejected at the TCP level using TCP wrappers. You can limit client-host access through TCP wrapping. This enables you to have non host-based protection for initial TCP connections to a Directory Server.

Although you can set TCP wrapping for Directory Server, TCP wrapping can result in significant performance degradation, especially during a Denial of Service attack. The best performance is achieved by using a host-based firewall that is maintained outside Directory Server, or IP port filtering.

▼ To Enable TCP Wrapping

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Create a `hosts.allow` file or a `hosts.deny` file, somewhere within the instance path.**

   For example, create the file in *instance-path*`/config`. Ensure that the formatting of the files that you create comply with `hosts_access(4)`.

2. **Set the path to the access file.**

   ```
   $ dsconf set-server-prop -h host -p port host-access-dir-path:path-to-file
   ```

   For example:

   ```
   $ dsconf set-server-prop -h host -p port \
   host-access-dir-path:/local/ds1/config
   "host-access-dir-path" property has been set to "/local/ds1/config".
   The "/local/ds1/config" directory on host1 must contain valid hosts.allow
   and/or hosts.deny files.
   Directory Server must be restarted for changes to take effect.
   ```

▼ To Disable TCP Wrapping

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Set the host access path to "".**

  ```
  $ dsconf set-server-prop -h host -p port host-access-dir-path:""
  ```

◆ ◆ ◆ CHAPTER 7

Directory Server Password Policy

When a user connects to Directory Server, the user is authenticated. The directory can grant access rights and resource limits to the user depending on the identity established during authentication. An *account* in this chapter refers loosely to a user entry. The account also reflects the permissions for the user to perform operations on the directory. In this discussion of password policy, every account is associated with a user entry, and a password.

This chapter also addresses account activation, an aspect of password policy. The Directory Administrator can directly lock and unlock accounts, independently of password policy.

This chapter does not cover authentication methods. Some authentication methods, such as SASL GSSAPI and client SSL certificate-based authentication, do not involve the use of passwords. The information about password policy in this chapter does not apply to such authentication methods. See Chapter 5, "Directory Server Security," for instructions on configuring authentication mechanisms.

This chapter also does not cover the compatibility of password policies between Directory Server 7.0 and previous Directory Server versions. When you create a Directory Server 7.0 instance, the password policy implementation defaults to a Directory Server 5 compatible mode to facilitate upgrading from earlier versions. To take full advantage of the password policy features described in this chapter, you will need to change the password policy compatibility mode.

Caution – The `DS5–compatibility-mode` password policy is deprecated. You must switch to `DS6–mode` password policy in this version.

For more information about setting the password compatibility mode, see "Password Policy Compatibility" on page 215.

This chapter covers the following topics:

Password Policies and Worksheet

This section explains password policy settings and provides a worksheet to help you define password policies that fit your requirements.

Note – To use the default password policy, see "Managing the Default Password Policy" on page 195.

Password Policy Settings

When you specify a password policy in Directory Server, you either modify or create an entry that includes the object class `pwdPolicy(5dsoc)`.

When defining a password policy for a particular type of user, you need to consider the following:

- How accounts get locked out when an intruder appears to be trying to crack a password.

  See "Policy for Account Lockout" on page 191 for details.
- How password changes can be made.

  See "Policy for Password Changes" on page 192 for details.
- What password values are allowed.

  See "Policy for Password Content" on page 192 for details.
- How password expiration is handled.

  See "Policy for Password Expiration" on page 193 for details.
- If the server records the time of the last successful authentication.

  See "Policy for Tracking Last Authentication Time" on page 194.

Note – Password policies measure password length by the number of bytes, so a password containing multi-byte characters can meet password-length policy even if the password contains fewer characters than the policy's specified minimum. For example, a 7-character password with one 2-byte character satisfies a password policy with password minimum length set to 8.

Subsequent sections in this chapter explain how you handle these areas of password policy. Use the "Worksheet for Defining Password Policy" on page 194 to clarify each password policy that you plan to implement.

Policy for Account Lockout

This section explains the policy attributes that govern account lockout.

A Directory Server account refers loosely to a user's entry and to the permissions that user has to perform operations on the directory. Each account is associated with a bind DN and a user password. When an intruder appears to be trying to crack a password, you want Directory Server to lock the account. The lock prevents the intruder from using the account to bind. The lock also prevents the intruder from being able to continue the attack.

As administrator, you can also manually render inactive an account or the accounts of all users who share a role. See "Manually Locking Accounts" on page 214 for instructions. Yet, a key part of your password policy is specifying under what circumstances Directory Server locks an account *without* your intervention.

First of all, you must specify that Directory Server can use `pwdLockout(5dsat)` to automatically lock accounts when too many failed binds occur. Directory Server keeps track of consecutive failed attempts to bind to an account. You use `pwdMaxFailure(5dsat)` to specify how many consecutive failures are allowed before Directory Server locks the account.

Directory Server locks accounts strictly according to password policy. The operation is purely mechanical. Accounts can lock not because an intruder is mounting an attack against the account, but because the user typed the password incorrectly. Thus, you can use `pwdFailureCountInterval(5dsat)` to specify how long Directory Server should wait between tries before cleaning out the records of failed attempts. You use `pwdLockoutDuration(5dsat)` to specify how long lockout should last before Directory Server automatically unlocks the account. The administrator does not have to intervene to unlock accounts of users who make legitimate mistakes with no malicious intent.

If your user data is replicated across a replication topology, lockout attributes are replicated along with other entry data. The `pwdIsLockoutPrioritized(5dsat)` attribute's default setting is `TRUE`, so updates for lockout attributes are replicated with a higher priority. A user is thus limited to making `pwdMaxFailure` consecutive failed attempts to bind to any single replica before being locked out, and probably fewer attempts at other replicas before being locked out.

For details about how to make sure that a user gets exactly pwdMaxFailure attempts before being locked out across an entire replicated topology, see "Preventing Authentication by Using Global Account Lockout" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

Policy for Password Changes

This section explains the policy attributes that govern changes to passwords.

In many deployments, Directory Server is the repository for identity data. Users should be able to change their own passwords, as specified by pwdAllowUserChange(5dsat), so you do not have to change the passwords.

After you allow users to change their own passwords, you might also want to control the circumstances under which users can change their passwords. You can use pwdSafeModify(5dsat) to specify that users who change a password must provide the correct existing password before they are allowed to replace the password. See "Modifying Passwords From the Command Line When pwdSafeModify Is TRUE" on page 209 for an example of how to modify the password. You can prevent users from reusing passwords by using pwdInHistory(5dsat) to specify how many passwords Directory Server remembers. You can also prevent users from changing their passwords too often by setting pwdMinAge(5dsat).

In many cases either you as administrator or some application that you manage creates user entries in the directory. You can assign a user password value to change when the user first binds to the new account. You might also have to reset a user password, after which the user should change the password when next using the account. Directory Server has a specific attribute, pwdMustChange(5dsat), that you can use to indicate whether a user must change passwords after the password value is reset by another user.

You can also specify that the Directory Administrator is not subject to policy when changing passwords by setting passwordRootdnMayBypassModsChecks(5dsat).

Policy for Password Content

This section explains the policy attributes that govern password content.

Although password values are not generally returned in directory searches, an attacker could potentially gain access to the directory database. Therefore, password values are generally stored in one of the supported hashed formats that you specify using passwordStorageScheme(5dsat).

You can also enforce a check that passwords meet your definition of minimum password quality, by setting pwdCheckQuality(5dsat). The server then checks that the password does not match any of the values of the cn, givenName, mail, ou, sn, or uid attributes. The comparison of the password with any of these attributes is case-insensitive.

Additional checks are available with pwdCheckQuality(5dsat) set. You can enforce that passwords have at least a specified number of characters by setting pwdMinLength(5dsat). Furthermore, when the Strong Password Check plug-in is enabled, Directory Server checks that the password does not contain strings from the dictionary file that the plug-in uses. The server also checks that the password contains an appropriate mix of different types of characters.

You can enable strong password checking with the `dsconf set-server-prop` command. Use the `pwd-strong-check-enabled` property to turn on the plug-in, and restart the server for the change to take effect. Use the `pwd-strong-check-require-charset` property to specify what character sets to require in passwords. The `pwd-strong-check-require-charset` property takes a mask of the following values:

`lower`	The new password must include a lower case character.
`upper`	The new password must include an upper case character.
`digit`	The new password must include a digit.
`special`	The new password must include a special character.
`any-two`	The new password must include at least one character from each of at least two of the above mentioned character sets.
`any-three`	The new password must include at least one character from each of at least three of the above mentioned character sets.

The default setting for the `pwd-strong-check-require-charset` property is `lower && upper && digit && special`.

Policy for Password Expiration

This section explains the policy attributes that govern password expiration.

To ensure that users change their passwords regularly, you can configure Directory Server to have passwords expire after the passwords reach a certain age, by setting pwdMaxAge(5dsat).

Users must be informed that their passwords are going to expire. You can configure Directory Server to return a warning that the password used to bind is going to expire. Use pwdExpireWarning(5dsat) to define how long before expiration that the warning should be returned when a client binds. *Notice that the client application gets the warning. The user does not get the warning directly.* Client applications must notify the end user when the applications receive the warning that the password is about to expire.

You can allow users one or more tries to bind with an expired password, by setting pwdGraceAuthNLimit(5dsat). Users who failed to change their passwords in time can thus still bind to change their passwords. Be aware that, when a user binds with a grace login, the user can perform any operation. A grace login works as if the password had not expired.

Directory Server updates the operational attribute pwdChangedTime(5dsat) every time that the password on the entry is modified. As a result, if you wait to enable password expiration, user passwords that have already aged expire immediately when you enable password expiration. Use warnings and grace logins if this behavior is not what you intend.

Policy for Tracking Last Authentication Time

This section covers the use of the password policy attribute pwdKeepLastAuthTime(5dsat).

When set, pwdKeepLastAuthTime causes Directory Server to track the time of the last successful bind every time that a user authenticates. The time is recorded on the pwdLastAuthTime(5dsat) operational attribute of the user's entry.

Because this behavior adds an update for each successful bind operation, the pwdKeepLastAuthTime feature is not activated by default. You must explicitly turn the feature on to use it in your deployment.

Worksheet for Defining Password Policy

This worksheet is designed to help you define a password policy to implement either through the command-line interface or using Directory Service Control Center (DSCC). Use one worksheet for each password policy.

After you record the DN of the password policy entry, record your decisions about settings for attributes in each policy area. Also record your rationale for those settings.

Password Policy Worksheet
Password Policy Entry Distinguished Name
dn: cn=

Policy Area	Attribute	Write Your Settings Here	Write Your Rationale for Settings Here
Account Lockout	pwdFailureCountInterval(5dsat)		
	pwdIsLockoutPrioritized(5dsat)		
	pwdLockout(5dsat)		
	pwdLockoutDuration(5dsat)		
	pwdMaxFailure(5dsat)		

Policy Area	Attribute	Write Your Settings Here	Write Your Rationale for Settings Here
Password Changes	passwordRootdnMayBypassModsChecks(5dsat)		
	pwdAllowUserChange(5dsat)		
	pwdInHistory(5dsat)		
	pwdMinAge(5dsat)		
	pwdMustChange(5dsat)		
	pwdSafeModify(5dsat)		
Password Content	passwordStorageScheme(5dsat)		
	pwdCheckQuality(5dsat)		
	pwdMinLength(5dsat)		
Password Expiration	pwdExpireWarning(5dsat)		
	pwdGraceAuthNLimit(5dsat)		
	pwdMaxAge(5dsat)		
Tracking Last Authentication Time	pwdKeepLastAuthTime(5dsat)		

Note – When the `pwdCheckQuality` attribute is set to 2, the server can perform additional checks. When the Password Check plug-in is also enabled, settings for the plug-in affect what checks are performed the on values of new passwords.

Managing the Default Password Policy

The default password policy applies to all users in the directory instance who do not have a specialized policy defined. However, the default password policy does not apply to the Directory Manager. See "Which Password Policy Applies" on page 199 for details on policy scope.

The default password policy is the one policy that you can configure using the `dsconf` command. You can also view default password policy by reading `cn=Password Policy,cn=config`.

This section shows the policy attributes for each policy area and the related `dsconf` server properties. It also explains how to view and change default password policy settings.

Correlation Between Password Policy Attributes and `dsconf` Server Properties

The following table shows the password policy attributes and related `dsconf` server properties for each password policy area.

Policy Area	Policy Attribute	`dsconf` Server Property
Account Lockout	`pwdFailureCountInterval`	`pwd-failure-count-interval`
	`pwdLockout`	`pwd-lockout-enabled`
	`pwdLockoutDuration`	`pwd-lockout-duration`
	`pwdMaxFailure`	`pwd-max-failure-count`
Password Changes	`passwordRootdnMayBypassModsChecks`	`pwd-root-dn-bypass-enabled`
	`pwdAllowUserChange`	`pwd-user-change-enabled`
	`pwdInHistory`	`pwd-max-history-count`
	`pwdMinAge`	`pwd-min-age`
	`pwdMustChange`	`pwd-must-change-enabled`
	`pwdSafeModify`	`pwd-safe-modify-enabled`
Password Content	`pwdCheckQuality`	`pwd-check-enabled`, `pwd-accept-hashed-password-enabled`, `pwd-strong-check-dictionary-path`, `pwd-strong-check-enabled`, `pwd-strong-check-require-charset`
	`pwdMinLength`	`pwd-min-length`
	`passwordStorageScheme`	`pwd-storage-scheme`
Password Expiration	`pwdExpireWarning`	`pwd-expire-warning-delay`
	`pwdGraceAuthNLimit`	`pwd-grace-login-limit`
	`pwdMaxAge`	`pwd-max-age`
Tracking Last Authentication Time	`pwdKeepLastAuthTime`	`pwd-keep-last-auth-time-enabled`

Note – The properties that correlate to `pwdCheckQuality` configure the Password Check plug-in. Therefore, the five properties apply to the entire server instance. The five properties thus also apply to other password policies where `pwdCheckQuality: 2`.

▼ To View Default Password Policy Settings

You can view default password policy settings with the `dsconf` command.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Read the default password policy configuration.**

```
$ dsconf get-server-prop -h host -p port -v -i \
-w password-file | grep ^pwd-
```

The *password-file* contains the password of directory manager.

```
pwd-accept-hashed-pwd-enabled      :  N/A
pwd-check-enabled                  :  off
pwd-compat-mode                    :  DS5-compatible-mode
pwd-expire-no-warning-enabled      :  on
pwd-expire-warning-delay           :  1d
pwd-failure-count-interval         :  10m
pwd-grace-login-limit              :  disabled
pwd-keep-last-auth-time-enabled    :  off
pwd-lockout-duration               :  1h
pwd-lockout-enabled                :  off
pwd-lockout-repl-priority-enabled  :  on
pwd-max-age                        :  disabled
pwd-max-failure-count              :  3
pwd-max-history-count              :  disabled
pwd-min-age                        :  disabled
pwd-min-length                     :  6
pwd-mod-gen-length                 :  6
pwd-must-change-enabled            :  off
pwd-root-dn-bypass-enabled         :  off
pwd-safe-modify-enabled            :  off
pwd-storage-scheme                 :  SSHA
pwd-strong-check-dictionary-path   :  instance-path/plugins/words-english-big.txt
pwd-strong-check-enabled           :  off
pwd-strong-check-require-charset   :  lower
pwd-strong-check-require-charset   :  upper
pwd-strong-check-require-charset   :  digit
pwd-strong-check-require-charset   :  special
pwd-supported-storage-scheme       :  CRYPT
```

```
pwd-supported-storage-scheme       :  SHA
pwd-supported-storage-scheme       :  SSHA
pwd-supported-storage-scheme       :  NS-MTA-MD5
pwd-supported-storage-scheme       :  CLEAR
pwd-user-change-enabled            :  on
```

▼ To Change Default Password Policy Settings

You can change the default password policy by setting server properties with the `dsconf` command.

Note – Before completing this procedure, read and complete the "Worksheet for Defining Password Policy" on page 194.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Translate the settings from your worksheet into `dsconf` command property settings.**

2. **Use the `dsconf set-server-prop` command to change default password policy properties appropriately.**

   For example, the following command allows the Directory Manager to violate the default policy when modifying passwords:

   ```
   $ dsconf set-server-prop -h host -p port pwd-root-dn-bypass-enabled:on
   ```

   The following command enables the policy that requires changing the password after a reset:

   ```
   # dsconf set-server-prop  -p 20390 pwd-must-change-enabled:on
   ```

Preventing Binds With No Password

Directory Server prevents authentication with a null password. All non-anonymous binds must therefore specify a password to bind to the directory. Otherwise, Directory Server returns an authentication error, `LDAP_INAPPROPRIATE_AUTH`.

You can disable this feature by setting the server property `require-bind-pwd-enabled` to `off` using the `dsconf set-server-prop` command.

The default value of the Require Bind on Authentication feature is `on`. Check this by using the following command:

```
# dsconf get-server-prop -p 20390 -w /tmp/.pwd-file require-bind-pwd-enabled
require-bind-pwd-enabled  :  on
```

Authenticating with a null password results in the following error message:

```
# ldapsearch -D cn=altrootdn -w '' -p 20390 -b cn=config 'objectclass=*' dn
ldap_simple_bind: Inappropriate authentication
ldap_simple_bind: additional info: binds with a dn require a password
```

Note that this feature does not block anonymous binds:

```
# ldapsearch -p 20390 -b cn=config 'objectclass=*' dn
version: 1
dn: cn=SNMP,cn=config
```

Disable this feature by setting it to `off`:

```
# dsconf set-server-prop -p 20390 -w /tmp/.pwd-file require-bind-pwd-enabled:off
# dsconf get-server-prop -p 20390 -w /tmp/.pwd-file require-bind-pwd-enabled
require-bind-pwd-enabled  :  off
```

This time authenticating with a null password succeeds:

```
# ldapsearch -D cn=altrootdn -w '' -p 20390 -b cn=config 'objectclass=*' dn
version: 1
dn: cn=SNMP,cn=config
```

Managing Specialized Password Policies

Specialized password policies are defined in a `pwdPolicy(5dsoc)` entry. A policy can be defined anywhere in the directory tree, typically in a subtree that is replicated with the accounts that the policy governs. The policy has a DN of the form `cn=`*policy name*`,`*subtree*.

After defining the password policy, you assign the password policy by setting the `pwdPolicySubentry(5dsat)` attribute in the desired user entry.

This section covers these topics:

Which Password Policy Applies

Directory Server allows you to configure multiple password policies. This section explains default password policies and specialized password policies. This section also explains which policy is enforced when multiple password policies could apply to a given account.

When you first create a Directory Server instance, that instance has a default password policy. That default password policy is expressed in the configuration entry `cn=PasswordPolicy,cn=config`. The default password policy applies to all accounts in the directory except for the Directory Manager.

As in all Directory Server password policies, `cn=PasswordPolicy,cn=config` has object class `pwdPolicy(5dsoc)` and object class `sunPwdPolicy(5dsoc)`.

Note – When you create a Directory Server instance, password policy attributes remain in Directory Server 5 compatible mode to facilitate upgrading from earlier versions. In Directory Server 5 compatible mode, Directory Server also handles password policy entries that have object class `passwordPolicy(5dsoc)`.

After your upgrade is complete, you use the new password policy in fully featured mode, as described in *Sun Directory Server Enterprise Edition 7.0 Upgrade and Migration Guide*. The administrative move is transparent to directory applications.

This chapter covers password policy configuration using the new password policy features.

You can change the default password policy to override the default settings. You can use the `dsconf(1M)` command to set the server properties for default password policy. Such server property names typically start with the `pwd-` prefix. When changing settings for such properties, you override the default password policy for the instance. Replication does not, however, copy the changes to replicas. The changes that you make to the default password policy are part of the configuration for the instance, not part of the directory data.

In addition to configuring the default password policy, you can also configure *specialized password policies*. A specialized password policy is defined by an entry in the directory tree. The specialized password policy entry has the same object class, `pwdPolicy(5dsoc)`, as the default password policy, and therefore takes the same policy attributes. Because specialized password policies are regular directory entries, policy entries are replicated in the same manner as regular directory entries.

A user entry references a specialized password policy through the value of the operational attribute `pwdPolicySubentry(5dsat)`. When referenced by a user entry, a specialized password policy overrides the default password policy for the instance. In many deployments, you assign users roles. You can configure roles to work with class of service (CoS) to determine the password policies that apply to user accounts, by setting the `pwdPolicySubentry` value. To override the password policy set by a role, change the `pwdPolicySubentry` value on that user's entry directly.

To summarize this section, initially the default password policy applies. You can change the default password policy to override the defaults. You can then create specialized password policy entries to override the default password policy. When you assign password policy with roles and CoS, you can override the CoS-assigned policy by specifying a password policy for an individual entry.

▼ To Create a Password Policy

You create and modify specialized password policies in the same way that you create and modify any other directory entry. The following procedure demonstrates use of a text editor to write the password policy entry in LDIF. Then you use the `ldapmodify` command with the `-a` option to add the password policy entry to the directory.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Example data as shown here is from `Example.ldif` unless stated otherwise.

1 Complete a password policy worksheet for the policy you want to create.

See "Worksheet for Defining Password Policy" on page 194 for a sample.

2 Write a password policy entry, in LDIF, that is based on the worksheet.

For example, the following policy entry specifies a password policy for temporary employees at Example.com, whose subtree root is `dc=example,dc=com`:

```
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: sunPwdPolicy
objectClass: LDAPsubentry
cn: TempPolicy
pwdAttribute: userPassword
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
```

In addition to the default password policy settings, the policy as shown here specifies additional behaviors. Password quality checks are enforced. Accounts are locked for five minutes, 300 seconds, after three consecutive bind failures. Passwords must be changed after the passwords are reset. After the policy is assigned to user accounts, the settings *explicitly* specified here override the default password policy.

3 Add the password policy entry to the directory.

For example, the following command adds the password policy for temporary employees at Example.com under `dc=example,dc=com`. The password policy has been saved in a file named `pwp.ldif`.

```
$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
adding new entry cn=TempPolicy,dc=example,dc=com
```

```
$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w --b dc=example,dc=com \
"(&(objectclass=ldapsubentry)(cn=temppolicy))"
Enter bind password:
version: 1
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: LDAPsubentry
cn: TempPolicy
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
$
```

As shown in `Example.ldif`, `kvaughan` is an Human Resources manager who has access to modify `dc=example,dc=com` entries. Vaughan's bind password, as shown in `Example.ldif`, is `bribery`.

See Also To define which user accounts are governed by the policies you define, see "To Assign a Password Policy to an Individual Account" on page 202 or "To Assign a Password Policy Using Roles and CoS" on page 203.

▼ To Assign a Password Policy to an Individual Account

This procedure assigns an existing password policy to a single user account.

Note – To complete this procedure, you must have a specialized password policy to assign. See "To Create a Password Policy" on page 201.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Example data shown here is from `Example.ldif` unless stated otherwise.

- **Add the password policy DN to the values of the `pwdPolicySubentry` attribute of the user entry.**

  For example, the following commands assign the password policy that is defined in "To Create a Password Policy" on page 201 to David Miller's entry, whose DN is `uid=dmiller,ou=people,dc=example,dc=com`:

```
$ cat pwp.ldif
dn: uid=dmiller,ou=people,dc=example,dc=com
changetype: modify
```

```
add: pwdPolicySubentry
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com

$ ldapmodify -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
modifying entry uid=dmiller,ou=people,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - -b dc=example,dc=com \
"(uid=dmiller)" pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=dmiller, ou=People, dc=example,dc=com
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com
$
```

As shown in `Example.ldif`, `kvaughan` is a Human Resources manager who has access to modify `dc=example,dc=com` entries. Vaughan's bind password, as shown in `Example.ldif`, is `bribery`.

▼ To Assign a Password Policy Using Roles and CoS

This procedure assigns an existing specialized password policy to a set of users by applying roles and class of service (CoS). See Chapter 9, "Directory Server Groups, Roles, and CoS," for more information about roles and CoS.

Note – To complete this procedure, you must have a specialized password policy to assign. See "To Create a Password Policy" on page 201.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Example data shown here is from `Example.ldif` unless stated otherwise.

1 Create a role for the entries to be governed by the password policy.

For example, the following commands create a filtered role for temporary employees at Example.com:

```
$ cat tmp.ldif
dn: cn=TempFilter,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: TempFilter
nsRoleFilter: (&(objectclass=person)(status=contractor))
```

```
description: filtered role for temporary employees

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f tmp.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com

$
```

As shown in `Example.ldif`, `kvaughan` is a Human Resources manager who has access to modify `dc=example,dc=com` entries. Vaughan's bind password, as shown in `Example.ldif`, is `bribery`.

2 Create a class of service to generate the DN of the password policy entry.

The DN is the value of the `pwdPolicySubentry` attribute of users who have the role that you created.

For example, the following commands create a filtered role for temporary employees at Example.com. The commands assign `cn=TempPolicy,dc=example,dc=com` to users who have the role.

```
$ cat cos.ldif
dn: cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: nsContainer

dn: cn="cn=TempFilter,ou=people,dc=example,dc=com",
 cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: LDAPsubentry
objectclass: costemplate
cosPriority: 1
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com

dn: cn=PolCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDN: cn=PolTempl,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f cos.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com

$
```

Users whose status is `contractor` now become subject to the password policy `cn=TempPolicy,dc=example,dc=com`.

▼ To Set Up a First Login Password Policy

In many deployments, the password policy to apply for new accounts differs from the password policy to apply for established accounts. This section demonstrates a first login password policy. The policy gives users three days to use a newly created account, and set their new passwords before that account is locked. The policy is designed to work in the same way for users whose passwords have been reset.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Create a specialized password policy for newly created accounts.

For example, add a password policy entry that sets expiration time to three days, which is 259,200 seconds.

Note – The global password policy must set pwdMustChange(5dsat) to TRUE, that is, users must change their passwords when they first bind.

```
$ dsconf set-server-prop -p port-no pwd-must-change-enabled:true
```

```
$ cat firstLogin.ldif
dn: cn=First Login,dc=example,dc=com
objectClass: top
objectClass: LDAPsubentry
objectClass: pwdPolicy
objectClass: sunPwdPolicy
cn: First Login
passwordStorageScheme: SSHA
pwdAttribute: userPassword
pwdInHistory: 0
pwdExpireWarning: 86400
pwdLockout: TRUE
pwdMinLength: 6
pwdMaxFailure: 3
pwdMaxAge: 259200
pwdFailureCountInterval: 600
pwdAllowUserChange: TRUE
pwdLockoutDuration: 3600
pwdMinAge: 0
pwdCheckQuality: 2
pwdMustChange: TRUE

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f firstLogin.ldif
Enter bind password:
adding new entry cn=First Login,dc=example,dc=com
```

2 Create a role that includes all newly created accounts.

In creating this role, set up some way to distinguish newly created accounts from established accounts.

a. Define new accounts as accounts that have a `pwdReset`(5dsat) attribute set to `TRUE`.

When a user's password is changed by another user, such as a password administrator, `pwdReset` is set to `TRUE`.

b. Create the role that identifies new accounts.

For example, the following commands create a role for accounts whose passwords have been reset.

```
$ cat newRole.ldif
dn: cn=First Login Role,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: First Login Role
nsRoleFilter: (pwdReset=TRUE)
description: Role to assign password policy for new and reset accounts

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f newRole.ldif
Enter bind password:
adding new entry cn=First Login Role,ou=people,dc=example,dc=com
```

3 Assign the password policy for newly created accounts with class of service.

```
$ cat newCoS.ldif
dn: cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: nsContainer

dn: cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: extensibleObject
objectClass: LDAPSubEntry
objectClass: CoSTemplate
cosPriority: 1
pwdPolicySubentry: cn=First Login,dc=example,dc=com

dn: cn=First Login CoS,dc=example,dc=com
objectClass: top
objectClass: LDAPSubEntry
objectClass: CoSSuperDefinition
objectClass: CoSClassicDefinition
```

```
cosTemplateDN: cn=First Login Template,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -f newCoS.ldif
Enter bind password:
adding new entry cn=First Login Template,dc=example,dc=com

adding new entry cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com

adding new entry cn=First Login CoS,dc=example,dc=com
```

Example 7–1 Checking Password Policy Assignment

Add a new user that fits the role that you have added. You add the user to verify that new users are subject to the new password policy, but existing users are not.

```
$ cat quentin.ldif
dn: uid=qcubbins,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: qcubbins
givenName: Quentin
sn: Cubbins
cn: Quentin Cubbins
mail: quentin.cubbins@example.com
userPassword: ch4ngeM3!
description: New account

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f quentin.ldif
Enter bind password:
adding new entry uid=qcubbins,ou=People,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - \
-b dc=example,dc=com uid=qcubbins nsrole pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=qcubbins,ou=People,dc=example,dc=com
nsrole: cn=first login role,ou=people,dc=example,dc=com
pwdPolicySubentry: cn=First Login,dc=example,dc=com
$ ldapsearch -b dc=example,dc=com uid=bjensen nsrole pwdPolicySubentry
version: 1
dn: uid=bjensen, ou=People, dc=example,dc=com
```

Notice that Barbara Jensen's existing account is governed by the default password policy. Quentin Cubbins's new account is governed, however, by the password policy that you defined.

Check the applied password policy settings by typing the following command:

```
# ldapsearch -D "cn=directory manager" -w - -b "cn=Password Policy,cn=config" -s base \
'(&(objectClass=ldapsubentry)(cn=Password Policy))'

version: 1
dn: cn=Password Policy,cn=config
objectClass: top
objectClass: ldapsubentry
objectClass: pwdPolicy
objectClass: sunPwdPolicy
objectClass: passwordPolicy
cn: Password Policy
pwdAttribute: userPassword
passwordStorageScheme: SSHA
passwordChange: on
pwdAllowUserChange: TRUE
pwdSafeModify: FALSE
passwordRootdnMayBypassModsChecks: off
passwordNonRootMayResetUserpwd: on
passwordInHistory: 0
pwdInHistory: 0
passwordMinAge: 0
pwdMinAge: 0
passwordCheckSyntax: off
pwdCheckQuality: 0
passwordMinLength: 6
pwdMinLength: 6
passwordMustChange: on
pwdMustChange: TRUE
passwordExp: off
passwordMaxAge: 0
pwdMaxAge: 0
passwordWarning: 86400
pwdExpireWarning: 86400
passwordExpireWithoutWarning: on
pwdGraceAuthNLimit: 0
pwdKeepLastAuthTime: FALSE
passwordLockout: off
pwdLockout: FALSE
passwordMaxFailure: 3
pwdMaxFailure: 3
passwordResetFailureCount: 600
pwdFailureCountInterval: 600
pwdIsLockoutPrioritized: TRUE
```

```
passwordUnlock: on
passwordLockoutDuration: 3600
pwdLockoutDuration: 3600
```

Modifying Passwords From the Command Line When `pwdSafeModify` Is `TRUE`

When the password policy for a user has `pwdSafeModify` set to `TRUE`, the old password must be provided with the new password to change the password. The command `dsconf set-server-prop pwd-safe-modify-enabled:on` has the same effect for the default password policy.

You can use the `ldappasswd(1)` command to change the password. This command provides support for safe password modification. This command implements RFC 3062, *LDAP Password Modify Extended Operation* (`http://www.ietf.org/rfc/rfc3062.txt`)

The following command lets Barbara Jensen change her own user password, connecting over simple authentication:

```
$ ./ldappasswd -h host -D uid=bjensen,ou=people,dc=example,dc=com \
-j old.pwd -T new.pwd -t old.pwd uid=bjensen,ou=people,dc=example,dc=com

ldappasswd: password successfully changed
```

You can also use the LDAP password modify extended operation. Setting up support for the extended operation is explained in "To Reset a Password With the Password Modify Extended Operation" on page 210.

Resetting Expired Passwords

When password policy enforces password expiration, some users will not change their passwords in time. This section shows how you can change passwords that have expired.

Note – Directory Server updates the operational attribute `pwdChangedTime(5dsat)` every time that the password on the entry is modified. As a result, if you wait to enable password expiration, user passwords that have already aged expire immediately when you enable password expiration. Use warnings and grace logins if this behavior is not what you intend.

This section includes procedures for resetting a password with the password modify extended operation and for allowing grace authentications when passwords expire.

The mechanisms described in this section are intended for use by administrators, or by applications that handle the actual user interaction with the directory. You typically rely on an application to ensure that the end user is in fact using the mechanisms in the way you intended.

▼ To Reset a Password With the Password Modify Extended Operation

User accounts are locked when passwords expire. When you reset the password, you unlock the account. The password can be reset by another user such as an administrator. After password reset, Directory Server unlocks the user account. Directory Server provides support for RFC 3062, *LDAP Password Modify Extended Operation* (`http://www.ietf.org/rfc/rfc3062.txt`). The extended operation enables you to allow a directory administrator or a directory application to unlock accounts through password reset.

Be cautious when allowing use of the password modify extended operation, as shown in this procedure. Limit access to administrators and applications that you trust. Do not allow passwords to travel over the network in clear text.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Give users access to a password administrator or to a password administration application.

2 Allow the password administrator access to use the password modify extended operation.

The following commands set an ACI to allow members of a `Password Managers` role to use the password modify extended operation when connected over SSL:

```
$ cat exop.ldif
dn: oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 1.3.6.1.4.1.4203.1.11.1
cn: Password Modify Extended Operation
aci: (targetattr != "aci")
(version 3.0; acl "Password Modify Extended Operation";
allow( read, search, compare, proxy )
(roledn = "ldap:///cn=Password Managers,dc=example,dc=com" and authmethod = "SSL");)

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f exop.ldif
Enter bind password:
adding new entry oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config

$
```

The entry under `cn=features,cn=config` allows you to manage access to operations that use the password modify extended operation.

3 **Have the password administrator reset the user password.**

This step unlocks the user account, and can be completed with the `ldappasswd(1)` command.

4 **(Optional) If the user must change the password, have the password administrator notify the user.**

Users must change their passwords after reset if the password policy that governs their entries includes `pwdMustChange: TRUE`.

▼ To Allow Grace Authentications When Passwords Expire

This procedure describes how to give users grace authentications, allowing users to change passwords that have expired.

The grace authentications are intended to be managed by an application that handles password policy request and response controls. The procedure shows a simple example of how to use the control in an application.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Make sure that users have access to an application that uses password policy request and response controls.**

The application should ensure that users handle grace authentications properly.

2 **Allow the application to use the password policy controls.**

The following commands set an ACI to allow members of a `Password Managers` role to use the password policy controls:

```
$ cat ctrl.ldif
dn: oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 1.3.6.1.4.1.42.2.27.8.5.1
cn: Password Policy Controls
aci: (targetattr != "aci")
(version 3.0; acl "Password Policy Controls";
allow( read, search, compare, proxy )
roledn = "ldap:///cn=Password Managers,dc=example,dc=com";)

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f ctrl.ldif
Enter bind password:
adding new entry oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config

$
```

The entry under `cn=features,cn=config` has the sole purpose of allowing you to manage access to operations that use the password policy request and response controls.

3. **Set `pwdGraceAuthNLimit` in the password policy to the number of authentications to allow after the password has expired.**

4. **Make sure that the application guides the end user to change the expired password promptly before grace authentications are exhausted.**

Caution – The `DS5–compatibility-mode` password policy is deprecated. You must switch to `DS6–mode` password policy in this version.

Setting Account Properties

The following sections describe how to set the look-through limit, size limit, time limit and idle timeout of an account.

▼ To Set the Look-Through Limit for an Account

- **Use the `ldapmodify` command to set the value of `nsLookThroughLimit`.**

  The following command removes the look-through limit for Barbara Jensen:

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsLookThroughLimit
nsLookThroughLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ To Set the Size Limit for an Account

- **Use the `ldapmodify` command to set the value of `nsSizeLimit`.**

  The following command removes the size limit for Barbara Jensen:

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
```

```
changetype: modify
add: nsSizeLimit
nsSizeLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

To Set the Time Limit for an Account

- **Use the `ldapmodify` command to set the value of `nsTimeLimit`.**

  The following command removes the time limit for Barbara Jensen:

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsTimeLimit
nsTimeLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

To Set the Idle Timeout for an Account

- **Use the `ldapmodify` command to set the value of `nsIdleTimeout`.**

  The following command sets the idle timeout for Barbara Jensen to five minutes (300 seconds):

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsIdleTimeout
nsIdleTimeout: 300
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
```

Manually Locking Accounts

Directory Server allows you to configure password policy to force the lockout of accounts after a specified number of failed bind attempts. See "Policy for Account Lockout" on page 191 for details. This section covers manual account locking and activation tools that the Directory Manager can use.

The Directory Manager can manage account lockout without using the lockout duration timer. The locked account remains locked until the account is explicitly activated. The Directory Manager can also render certain accounts inactive for an indefinite period of time.

This section shows how to check account status, render accounts inactive, and reactivate accounts.

▼ To Check Account Status

Check account status as shown here.

Note – You must bind as the Directory Manager.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Use the `dsutil account-status` command to check the status the account or role.**

  The following command checks Barbara Jensen's account status:

  ```
  $ dsutil account-status -p port-number -w pwd.txt \
  uid=bjensen,ou=people,dc=example,dc=com

  uid=bjensen,ou=people,dc=example,dc=com activated.
  ```

  See the `dsutil(1M)` man page for details.

▼ To Render Accounts Inactive

Render an account or a role inactive as shown here.

Note – You must bind as the Directory Manager.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Use the `dsutil account-inactivate` command to render the account or role inactive.**

  The following command renders Barbara Jensen's account inactive:

  ```
  $ dsutil account-inactivate -p port-number -w pwd.txt \
  uid=bjensen,ou=people,dc=example,dc=com

  uid=bjensen,ou=people,dc=example,dc=com inactivated.
  $
  ```

  See the `dsutil(1M)` man page for details.

▼ To Reactivate Accounts

Unlock an account or a role as shown here.

Note – You must bind as the Directory Manager.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Use the `dsutil account-activate` command to reactivate the account or role.**

  The following command renders Barbara Jensen's account active again:

  ```
  $ dsutil account-activate -p port-number -w pwd.txt \
  uid=bjensen,ou=people,dc=example,dc=com

  uid=bjensen,ou=people,dc=example,dc=com activated.
  ```

  `dsutil(1M)` man page for details.

Password Policy Compatibility

For migration purposes, the new password policy maintains compatibility with previous Directory Server versions by implementing a compatibility mode. The compatibility mode determines whether password policy attributes are handled as *old* attributes or *new* attributes, where *old* refers to Directory Server 5.2 password policy attributes.

This section contains information to help you set the compatibility mode and to decide which mode is appropriate for your deployment.

Setting the Compatibility Mode

The compatibility mode can be read using `dsconf` command as follows:

```
$ dsconf get-server-prop pwd-compat-mode
```

The `pwd-compat-mode` property can have one of the following values:

`DS5-compatible-mode` The purpose of `DS5-compatible-mode` is to allow an existing replicated topology to be migrated from Directory Server 5.2 instances to Directory Server 7.0 instances. A Directory Server 7.0 instance in `DS5-compatible-mode` accepts updates containing either old or new password policy attributes, and produces updates containing both sets of attributes. Updates can arrive from an LDAP client or from replication, and changes are produced locally as a result of password policy evaluation (for example, as a result of a failed authentication or a password change). Note that any version 5.2 instance consuming replicated updates produced by a version 7.0 instance (either directly or through another instance) must have its schema updated with `00ds6pwp.ldif` as described in "Issues With the Password Policy" in *Sun Directory Server Enterprise Edition 7.0 Upgrade and Migration Guide*. While the Directory Server 5.2 instance will ignore any new attributes during password policy processing, when an entry containing the new attributes is modified at that instance, without the schema update, the schema check will fail when the modified entry is written.

`DS6-migration-mode` `DS6-migration-mode` is an intermediate step between `DS5-compatible-mode` and `DS6-mode`, All policy decisions are based on the new password policy attributes and the old (Directory Server 5.2) password policy attributes are removed from the server's data. Since the number of policy configuration entries is small, the old password policy configuration attributes are cleaned from all policy entries as soon as the instance is advanced to `DS6-migration-mode`. However, the cleanup of a user entry is deferred until the entry is modified as part of normal password policy processing during a password modify operation. This approach allows the cleanup to proceed gradually, so as to not degrade the server's performance. The modifications necessary to remove any old policy attributes are not replicated so that they do not interfere with the operation of instances still in `DS5-compatible-mode`.

`DS6-mode` A Directory Server instance in `DS6-mode` uses only new policy attributes in computing password policy decisions. Any old password policy attributes remaining in an entry are ignored.

The compatibility mode is set using the `dsconf` command as follows:

```
$ dsconf pwd-compat new-mode
```

The *new-mode* action takes one of the following values:

`to-DS6-migration-mode`	Change to `DS6-migration-mode` from `DS5-compatible-mode`. Once the change is made, only `DS6-migration-mode` and `DS6-mode` are available.
`to-DS6-mode`	Change to `DS6-mode` from `DS6-migration-mode`. Once the change is made, only `DS6-mode` is available.

The server state can move only towards stricter compliance with the new password policy specifications.

Note – The `DS5–compatibility-mode` password policy is deprecated. You must switch to `DS6–mode` password policy in this version.

Guidelines for Choosing a Compatibility Mode

The `pwd-compat-mode` setting affects the internal server operation and is largely isolated from the password policy behavior seen by an LDAP client. In particular, the `pwd-compat-mode` setting does not affect the range of server responses to an LDAP client authentication (bind).

Note – The configuration and operational attributes used to implement the password policy depend on the `pwd-compat-mode` setting. Therefore, an LDAP application that accesses the old (Directory Server 5.2) attributes will need to be modified prior to advancing the `pwd-compat-mode` beyond the initial `DS5-compatible-mode`.

Note – `DS5-compatible-mode` is the default setting. If you upgrade an existing server to Directory Server 7.0 or if you create a new Directory Server 7.0 instance, the compatibility state is set to `DS5-compatible-mode`.

This section provides details about the compatibility mode appropriate to your Directory Server deployment.

New Directory Server 7.0 Deployment

If you install a standalone Directory Server instance or are deploying a new replicated topology, set the compatibility mode to `DS6-mode` to immediately take advantage of the features available in the new password policy implementation. Since a new Directory Server 7.0 instance is created with the compatibility mode set to `DS5-compatible-mode`, you will need to remember to advance the instance to `DS6-mode` before installing it into a replicated topology whose instances are already set to `DS6-mode`.

Migrating a Deployment to Directory Server 7.0

If you are migrating an existing replicated topology, as long as at least one Directory Server 5.2 instance remains in the replication topology, all of the Directory Server 7.0 instances must be set to `DS5-compatible-mode`.

Once a replicated topology has been completely migrated (in `DS5-compatible-mode`), you can consider advancing from maintaining compatibility with the old password policy to using the new password policy exclusively. Moving from `DS5-compatible-mode` to `DS6-mode` occurs in two phases, which includes an intermediate stage in `DS6-migration-mode`. First, the Directory Server 7.0 instances must be left in `DS5-compatible-mode` for an entire password expiration cycle so that the user entries are populated with the new `pwdChangedTime` attribute. Any applications that depend on the old password policy attributes must also be migrated to the new attributes while the Directory Server 7.0 instances are in `DS5-compatible-mode`, since the old attributes are no longer available in `DS6-migration-mode`. At this point, the instances comprising the replicated topology can be advanced to `DS6-migration-mode`.

The second phase consists of running all instances of the replicated topology in the intermediate `DS6-migration-mode` to clean out the old operational attributes in the entries. This cleanup must occur before advancing from `DS6-migration-mode` to `DS6-mode`. Otherwise, the stale attributes will remain in the entries. To mitigate the overhead of cleaning the old password policy operational attributes, the Directory Server 7.0 instance only removes the attributes in conjunction with a password modify. Thus a simple approach to the cleanup, assuming the password expiration feature is enabled, is to leave the Directory Server 7.0 instances in `DS6-migration-mode` for an entire password expiration cycle. Finally, once the old password policy attributes have been cleaned from the entries, the instances can be moved to `DS6-mode`. Remember that the new Directory Server 7.0 instance is created and set to `DS5-compatible-mode`. You will need to remember to advance the instance to `DS6-mode` before installing it into a replicated topology whose instances are already at `DS6-mode`.

The following table shows the allowed combinations of Directory Server versions and password policy compatibility modes. Note that at most two variations are allowed in a replicated topology at any time. For example, if a topology contains a Directory Server 7.0 instance in `DS5-compatible-mode` and one in `DS6-migration-mode`, then those are the only two variants allowed: no legacy Directory Serverinstances or Directory Server 7.0 instances in `DS6-mode` are allowed.

TABLE 7–1 Directory Server Password Policy Mode Interoperability

	Directory Server 5.2	**Directory Server 7.0 in** `DS5-compatible-mode`	**Directory Server 7.0 in** `DS6-migration-mode`	**Directory Server 7.0 in** `DS6-mode`
Directory Server 5.2	X	X		
Directory Server 7.0 in `DS5-compatible-mode`	X	X	X	
Directory Server 7.0 in `DS6-migration-mode`		X	X	X
Directory Server 7.0 in `DS6-mode`			X	X

Administrative Password Reset Classification

Password policy features such as must-change-on-reset (`pwd-must- change-enabled`) and administrative bypass of password quality checks (`pwd-root-dn-bypass-enabled`) depend on classifying the modification of the `userPassword` attribute as either a self-change or an administrative reset.

In Directory Server 5.2, by default, only the Directory Manager can perform an administrative reset of a user's password. Any other password change is considered as a self-change. Directory Server 5.2p4 introduced the password policy configuration attribute `passwordNonRootMayResetUserpwd` that, when enabled, limits the `userPassword` modify operations that are considered as a self-change to the following two cases:

1. A user is authenticated and changing the password of his or her own account.
2. An administrator changes the password, but the LDAP Proxied Authorization Control (`http://www.ietf.org/rfc/rfc4370.txt`) is set for the `userPassword` modify operation, and the proxied user DN is the target of the modify operation.

Any other password change is considered as an administrative reset. This feature eliminates the requirement of using Directory Manager for routine password administration, while the simple other-than- self (password change made by any other user but not by self) test avoids the complexity of a separate scheme to identify administrative users.

In this version, Directory Server evaluates password changes similar to Directory Server 5.2 with `passwordNonRootMayResetUserPassword` enabled. That is, Directory Server considers a password change as an administrative reset except for a user changing his or her own password, or when the proxied authorization control is used. Even though the `passwordNonRootMayResetUserpwd` attribute can be present in a Directory Server password policy configuration entry when the instance is in Directory Server 5.2 compatible mode, the attribute can not be modified and the feature is always enabled.

If your Directory Server 5.2 based LDAP application uses an administrative account other than Directory Manager to change a password on behalf of a user (that is, the change should be a self-change), when the application is used with Directory Server 7.0, the change will be considered as an administrative reset. You can restore the original behavior by using the LDAP Proxied Authorization Control (`http://www.ietf.org/rfc/rfc4370.txt`) with the `userPassword` modify operation. The proxied authorization control handles the operation as if it is invoked by the proxied user. The control is available in the LDAP C SDK (`http://wiki.mozilla.org/LDAP_C_SDK`) and LDAP SDK for Java (`http://www.mozilla.org/directory/javasdk.html`), and the `ldapmodify` command. Invoke the proxied authorization control using the `ldapmodify` command as follows:

```
$ ldapmodify -D <administrative-user-DN > -Y <proxied-user-DN>
```

Note – The `ldapmodify` commands from other products might use a different flag, or might not support the proxied authorization control at all.

◆ ◆ ◆ CHAPTER 8

Directory Server Backup and Restore

The data managed by your Directory Server is often imported in bulk. Directory Server Enterprise Edition provides tools for importing and exporting entire suffixes. It also provides tools for making backups of all suffixes at once and for restoring all data from a backup.

Before starting any backup or restore operation, ensure that you design a backup and restore strategy to suit your situation. For more information about the different backup options, considerations to take into account, and guidelines for planning a backup and restore strategy, see "Designing Backup and Restore Policies" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

This chapter covers the following topics:

Binary Backup

This section explains how to perform a binary backup of directory data. In addition to the binary backup procedures in this section, you can make a binary copy to use for initializing a suffix in a replication topology. See "Initializing a Replicated Suffix by Using Binary Copy" on page 271.

Backing Up Directory Data Only

A binary data backup saves a copy of your directory data that you can use if the database files later become corrupted or deleted. This operation takes the back up of the database only and

does not back up any other data such as configuration data and certificates. If you want to back up the whole Directory Server for disaster recovery, see "Disaster Recovery" on page 231.

Caution – The maximum period between two backups should not exceed the smaller of `repl-purge-delay` and `repl-cl-max-age`. The `repl-cl-max-age` property specifies the period of time, after which internal purge operations are performed on the change log. The change log maintains a record of updates, which might or might not have been replicated. Get the purge delay information using the following command:

```
dsconf get-suffix-prop -h host -p port suffix-DN repl-purge-delay repl-cl-max-age
```

If your backup is performed less frequently than the purge delay, the change log might be cleared before it has been backed up. Changes will therefore be lost if you use the backup to restore data.

The consumer server stores internal information about updates to the replicated suffix contents, and the purge delay parameter, `repl-purge-delay`, specifies how long it must keep this information. The purge delay determines in part how long replication between the consumer and its master can be interrupted and still recover normally. It is related to the `repl-cl-max-age` parameter of the change log on its supplier server. The shorter of these two parameters determines the longest time that replication between the two servers can be disabled or down and still recover normally. The default value of 7 days is sufficient in most cases.

All backup procedures described in this section store a copy of the server files on the same host by default. You should then copy and store your backups on a different machine or file system for greater security.

▼ To Back Up Your Directory Data

Your Directory Server must be stopped to run the `dsadm backup` command.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Back up your directory data.**

  ```
  $ dsadm backup [ -f FLAG ] ... INSTANCE_PATH ARCHIVE_DIR
  ```

  For example:

  ```
  $ dsadm backup /local/dsInst /local/tmp/20091005
  ```

Note – By default, binary backup commands will run a database recovery on the backup databases. See `dsadm(1M)` for disabling this behavior.

You can back up directory data while the server is running by using the command `dsconf backup` command. However, if changes are made to the directory data while the backup is running, proper recovery will take longer.

Never stop the server during a backup operation.

For more information about the `dsadm backup`, `dsconf backup` commands and backup flags, see the `dsadm(1M)` and `dsconf(1M)` man pages.

▼ To Back Up the `dse.ldif` File

When restoring a server, all the configuration data such as certificates, schema, and plugins must contain the same configuration information as when the server was backed up. The following task shows how to back up the `dse.ldif` file and the rest of the configuration information can be backed up in the same manner.

- **Back up your `dse.ldif` configuration file.**

  `$ cp` *instance-path*`/config/dse.ldif` *archive-dir*

  When you perform the following actions, Directory Server automatically backs up the `dse.ldif` configuration file in the directory *instance-path*`/config`.

  - When you start Directory Server, a backup of the `dse.ldif` file is created in a file named `dse.ldif.startOK`.
  - When you make modifications to the `cn=config` branch, the file is first backed up to a file named `dse.ldif.bak` in the `config` directory before the server writes the modifications to the `dse.ldif` file.

Backing Up a File System

This procedure optionally uses the *frozen mode* feature. Frozen mode enables you to stop database updates on disk so that a file system snapshot can be taken safely. You can use frozen mode as an additional measure for ensuring a robust backup.

If the server instance is stopped, *frozen mode* does not apply.

Your server must not write user data on the disk while the file system backup is in progress. If you are sure that no updates will occur during a certain time frame, make your backup during this time. If you cannot guarantee that there will be no updates, put your server into frozen mode before making a backup.

A server in frozen mode continues to write to the access and errors logs. In a single-server topology, operations received when frozen mode is on result in an LDAP error being returned. The error message logged is the standard error for the database being offline. In a replicated topology, a referral is returned. For frozen mode to work correctly, no other tasks should be running on the databases.

Note that the databases of a server in frozen mode are more stable than those in read-only mode. Unlike frozen mode, read-only mode permits tasks to be created and configuration entries to be modified. When frozen mode is on, all configured databases are taken offline. Any internal operations in progress are notified of the database going offline. LDAP operations in progress are completed, and the database environment is flushed. Subsequent incoming operations, including searches to user data, are refused until frozen mode is set to off. You can, however, search configuration parameters while frozen mode is on.

Frozen mode can be active only when the server is running. Restarting the server instance will also reset frozen mode to off.

To Back Up a File System

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1. **(Optional) Put your server into frozen mode.**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:frozen
   ```

2. **Back up your file system, using a tool appropriate to your file system type.**

3. **If your server is in frozen mode, make the server read-write again.**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:read-write
   ```

   If your server receives replication updates from another server, replication updates will start as soon as frozen mode is turned off.

To Restore the File System

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1. **Stop your server.**

   ```
   $ dsadm stop instance-path
   ```

2. **Restore your file system, using a tool appropriate to your file system type.**

3 **Start your server.**

```
$ dsadm start instance-path
```

Backing Up to LDIF

Backing up to LDIF allows you to back up directory data to a formatted LDIF file.

Exporting to LDIF

You can back up directory data by exporting the contents of a suffix in the LDIF format. Exporting data can be useful for doing the following:

- Backing up the data in your server
- Copying your data to another directory server
- Exporting your data to another application
- Repopulating suffixes after a change to your directory topology

The configuration information cannot be exported.

Caution – Do not stop the server while an export operation is in progress.

▼ To Export a Suffix to LDIF

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Use one of the following commands to export a suffix to an LDIF file:**
  - If your server is local and stopped, type:

    ```
    $ dsadm export instance-path suffix-DN LDIF-file
    ```

    The `dsadm export` takes only offline backups.
  - If your server is running (local or remote), type:

    ```
    $ dsconf export -h host -p port suffix-DN LDIF-file
    ```

    The `dsconf export` takes only online backups.

The following example uses `dsconf export` to export two suffixes to a single LDIF file:

```
$ dsconf export -h host1 -p 1389 ou=people,dc=example,dc=com \
 ou=contractors,dc=example,dc=com /local/dsInst/ldif/export123.ldif
```

The ldif file will be created on the machine running the server instance not on the machine running the `dsconf` command.

The `dsadm export` and `dsconf export` commands can also be used with the `--no-repl` option to specify that no replication information is to be exported. The default is that the replicated suffix is exported to an LDIF file with replication information. The resulting LDIF file will contain attribute subtypes that are used by the replication mechanism. This LDIF file can then be imported on the consumer server to initialize the consumer replica, as described in "Initializing Replicas" on page 267

For more information about these commands, see the `dsadm`(1M) and `dsconf`(1M) man pages.

Binary Restore

The following procedures describe how to restore suffixes in your directory. Your server must have been backed up using the procedures described in "Backing Up Directory Data Only" on page 221. Before restoring suffixes involved in replication agreements, read "Restoring Replicated Suffixes" on page 227.

Caution – Do not stop the server during a restore operation. Because restoring your server overwrites any existing database files, any modifications that were made to the data since the backup are lost.

▼ To Restore Your Server

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Use one of the following commands to restore your server:**
  - If your server is local and stopped, type:

    ```
    $ dsadm restore instance-path archive-dir
    ```

    For example, to restore a backup from a backup directory, type:

    ```
    $ dsadm restore /local/dsInst/ local/ds/bak/2006_07_01_11_34_00
    ```

  - If your server is running, type:

    ```
    $ dsconf restore -h host -p port archive-dir
    ```

    For example, to restore a backup from a backup directory:

    ```
    $ dsconf restore -h host1 -p 1389 /local/dsInst/bak/2006_07_01_11_34_00
    ```

    Do not stop the server during a restore operation.

Note – The backup copy must be saved on the server, not on the system that is running the `dsconf` command.

After a restore, there is no way to go back to the original server content.

Note – To save disk space, you can restore server by moving files in place of copying them. You can perform this operation by setting the `–f move-archive` flag with the `dsadm restore` or the `dsconf restore` command. However, if the transaction logs cannot be moved during the copyless restore operation, the copy operation is performed instead.

For this operation to complete successfully, the backup and instance files must be on the same filesystem. If you choose to perform the copyless restore, the server data is overwritten with the data in the backup copy and the backup copy also gets destroyed.

For more information about these commands, see the `dsadm`(1M) and `dsconf`(1M) man pages.

Restoring Replicated Suffixes

Suffixes that are replicated between supplier servers and consumer servers require special consideration before being restored. If possible, update the suffix through the replication mechanism instead of restoring it from a backup.

When you restore a supplier or hub instance, the server configuration must be the same as it was when the backup was made. To ensure this, restore the `dse.ldif` file before restoring Directory Server data. Use `--safe` option while starting the server. For more information, see "To Start, Stop, and Restart Directory Server" on page 46.

This section explains how and when to restore a replica, and how to ensure that it is synchronized with other replicas after the operation. To initialize a replica, see "Initializing Replicas" on page 267.

If you have a large replicated suffix and you want to add many entries and ensure that replication updates are added correctly, see "Incrementally Adding Many Entries to Large Replicated Suffixes" on page 274.

This section contains information about the following:

Restoring the Supplier in a Single-Master Scenario

A suffix that is a single-master supplier contains the authoritative data for the entire replication topology. Therefore, restoring this suffix is equivalent to reinitializing all data in the entire topology. You should restore a single master only if you want to reinitialize all data from the contents of the backup to be restored.

If the single-master data is not recoverable due to an error, consider using the data on one of the consumers because it might contain updates that are more recent than a backup. In this case, you need to export the data from the consumer replica to an LDIF file, and reinitialize the master from the LDIF file.

Whether you restore a backup or import an LDIF file on a master replica, you must then reinitialize all of the hubs and consumer replicas that receive updates from this replica. A message is logged to the supplier servers' log files to remind you that reinitialization of the consumers is required.

Restoring a Supplier in a Multi-Master Scenario

In multi-master replication, the other masters each contain an authoritative copy of the replicated data. You cannot restore an old backup because it might be out of date with the current replica contents. If possible, allow the replication mechanism to bring the master up to date from the contents of the other masters.

If that is not possible, restore a multi-master replica in one of the following ways:

- The simplest way is not to restore a backup, but to reinitialize the intended master from one of the other masters. This ensures that the latest data is sent to the intended master and that the data will be ready for replication. See "Replica Initialization From LDIF" on page 268.
- For replicas with millions of entries, it can be faster to make a binary copy to restore a more recent backup of one of the other masters. See "Initializing a Replicated Suffix by Using Binary Copy" on page 271.
- If you have a backup of your master that is not older than the maximum age of the change log contents on *any* of the other masters, the backup may be used to restore this master. See "To Modify Change Log Settings on a Master Replica" on page 260 for a description of change log age. When the old backup is restored, the other masters will use their change logs to update this master with all modifications that have been processed since the backup was saved.

Regardless of how you restore or reinitialize, the master replica will remain in read-only mode after the initialization. This behavior allows the replica to synchronize with the other masters, after which time you may allow write operations, as described in "Restoring a Master in a Multi-Master Scenario" on page 230.

The advantage of allowing all replicas to converge before allowing write operations on the restored or reinitialized master is that none of the hub or consumer servers will require reinitialization.

Restoring a Hub

This section applies only in situations where the replication mechanism cannot automatically bring a hub replica up to date. Such situations include if the database files become corrupted or if replication has been interrupted for too long. In these cases, you need to restore or reinitialize the hub replica in one of the following ways:

- The simplest way is not to restore a backup, but to reinitialize the hub from one of the master replicas. This ensures that the latest data is sent to the hub and that the data will be ready for replication. See "Initializing a Suffix" on page 52.
- For replicas with millions of entries, it can be faster to make a binary copy to restore a more recent backup taken from another hub replicated suffix. See "Initializing a Replicated Suffix by Using Binary Copy" on page 271. If there is no other hub replica to copy, reinitialize the hub as described in the previous item, or restore it as described in the next item, if possible.
- If you have a backup of your hub that is not older than the maximum age of the change log contents on *any* of its suppliers, either hub or master replicas, the backup may be used to restore this hub. When the hub is restored, its suppliers will use their change logs to update this hub with all modifications that have been processed since the backup was saved.

Note – Regardless of how you restore or reinitialize the hub replica, you *must* then reinitialize all consumers of this hub, including any other levels of hubs.

Restoring a Dedicated Consumer

This section applies only in situations where the replication mechanism cannot automatically bring a dedicated consumer replica up to date. Such situations include if the database files become corrupted or if replication has been interrupted for too long. In these cases, you need to restore or reinitialize the consumer in one of the following ways:

- The simplest way is not to restore a backup, but to reinitialize the consumer from one of its suppliers, either a master or a hub replica. This ensures that the latest data is sent to the consumer and that the data will be ready for replication. See "Replica Initialization From LDIF" on page 268.
- For replicas with millions of entries, it can be faster to make a binary copy to restore a more recent backup taken from another consumer replicated suffix. See "Initializing a Replicated Suffix by Using Binary Copy" on page 271. If there is no other consumer to copy, reinitialize the replica as described in the previous item or restore it as described in the next item, if possible.
- If the backup of your consumer is not older than the maximum age of change log contents on *any* of its suppliers, either hub or master replicas, the backup may be used to restore this consumer. When the consumer is restored, its suppliers will use their change logs to update the consumer with all modifications that have been processed since the backup was saved.

Restoring a Master in a Multi-Master Scenario

In the case of multi-master replication, other masters may process change operations while a given master is being restored. Therefore, when restoration is complete, the new master must also receive new updates that were not included in the restore data. Because restoring a master might take a significant amount of time, the number of pending updates might also be significant.

To allow convergence of these pending updates, newly restored masters are automatically set to read-only mode for client operations after restoration. This is true only when restoring a master by importing data from an LDIF file at the command line, or by using a backup to perform a binary copy.

Therefore, after restoration, a master in a multi-master configuration will process replication updates and allow read operations, but it will return referrals for all write operations from clients.

To verify that the new master is fully synchronized with the other masters before allowing updates, manually enable updates on an initialized master.

Note – With master replicas sending referrals because of this new behavior, clients wanting to perform write operations might reach their configured hop limit. You might need to increase the hop limit configuration for clients so they can reach an available master. If all master replicas are initialized or reinitialized, all write operations will fail because no replica will be accepting client updates.

In any case, monitor initialized masters closely, and set the referral attributes appropriately to maximize server response.

▼ To Begin Accepting Updates Through the Command Line

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

The following commands can be used in scripts that automate the process of initializing a multi-master replica.

1 **Use the `insync` tool to ensure that the replica has converged with all other masters.**

Replicas are in sync if the delay between modifications on all servers is zero or if the replica has never had any changes to replicate (delay of `-1`). For more information, see the `insync(1)` man page.

2 **Begin accepting updates.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN repl-accept-client-update-enabled:on
```

This command automatically sets the server to read-write mode.

Disaster Recovery

If you want to back up or restore your Directory Server for disaster recovery purposes, use the following procedures.

▼ To Make a Backup for Disaster Recovery

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1 **Make a backup of your database files by using the command `dsadm backup` or `dsconf backup`.**

Use the procedure in "Binary Backup" on page 221, and store the backup files in a safe place.

2 **Copy the configuration directory *instance-path*`/config` to a safe place.**

3. **Copy the schema directory** *instance-path*`/config/schema` **to a safe place.**

4. **Copy the alias directory** *instance-path*`/alias` **to a safe place.**

▼ To Restore for Disaster Recovery

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

Make sure you recover the system on the same hardware configuration. For example, the Solaris SPARC recovery mechanism should be used to restore Directory Server on Solaris SPARC.

1. **Install the same version of Directory Server that you had previously on the host.**

2. **Create a server instance by using the** `dsadm create` **command.**

   The `instance-path` must be the same as the original instance. See "Creating Suffixes" on page 47.

3. **Restore the configuration directory** *instance-path* `/config`**.**

4. **Restore the schema directory** *instance-path*`/config/schema.`

5. **Restore the alias directory** *instance-path*`/alias.`

6. **Ensure that the configuration for the restored server is correct.**

   For example, the directory structure and plug-in configuration must be the same as on the backed up server.

7. **Restore your database files by using the command** `dsconf restore`**.**

   Use the procedure in "Binary Restore" on page 226.

◆ ◆ ◆ CHAPTER 9

Directory Server Groups, Roles, and CoS

Beyond the hierarchical structure of data in a directory, managing entries that represent users often requires creating groups that share common attribute values. Directory Server provides advanced entry management functionality through groups, roles, and class of service (CoS).

This chapter covers the following topics:

About Groups, Roles, and Class of Service

Groups, roles, and CoS are defined as follows:

- Groups are entries that name other entries, either as a list of members or as a filter for members. For groups that consist of a list of members, Directory Server generates values for the `isMemberOf` attribute on each user entry. The `isMemberOf` attribute on a user entry thus shows all the groups to which that entry belongs.
- Roles provide the same functionality as groups, and more, through a mechanism that generates the `nsrole` attribute on each member of a role.
- CoS generates a computed attribute, which allows entries to share a common attribute value without having to store the attribute in each entry.

  You cannot use the `isMemberOf` attribute to make all the members of static groups automatically inherit from a common computed attribute value.

Directory Server provides the ability to perform searches that are based on the values of the roles, and groups and the CoS computed attributes. Filter strings used in any operation can include the `nsRole` attribute or any attribute generated by a CoS definition. Filter strings can

also be used to perform any of the comparison operations on the value of this attribute. However, computed CoS attributes cannot be indexed. Therefore, any search that involves a CoS-generated attribute might consume a large amount of resources in terms of time and memory.

To take full advantage of the features offered by roles, groups, and class of service, determine your grouping strategy in the planning phase of your directory deployment. Refer to Chapter 11, "Directory Server Groups and Roles," in *Sun Directory Server Enterprise Edition 7.0 Reference* for a description of these features and how they can simplify your topology.

To gain a deeper understanding of how roles and groups work, see Chapter 11, "Directory Server Groups and Roles," in *Sun Directory Server Enterprise Edition 7.0 Reference*. For a detailed description of CoS, see Chapter 12, "Directory Server Class of Service," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

Managing Groups

Groups enable you to associate entries for ease of administration. For example, using groups makes it easier to define access control instructions (ACIs). Group definitions are special entries that either name their members in a static list or provide a filter that defines a dynamic set of entries.

The scope of possible members of a group is the entire directory, regardless of where the group definition entries are located. To simplify administration, all group definition entries are usually stored in a single location, usually `ou=Groups` under the root suffix.

The two types of groups are static groups and dynamic groups.

- **Static groups.** The entry that defines a static group inherits from either the `groupOfNames` or `groupOfUniqueNames` object class. Group members are listed by their DN as multiple values of the `member` or `uniqueMember` attribute.

  Alternatively, you can use the `isMemberOf` attribute for static groups. The `isMemberOf` attribute is calculated and added to the user entry at the start of the search. It is then removed again after the search has finished. This functionality provides easy management of groups, and fast read access.
- **Dynamic groups.** The entry that defines a dynamic group inherits from the `groupOfURLs` object class. Group membership is defined by one or more filters that are specified in the multivalued `memberURL` attribute. The members in a dynamic group are the entries that match any one of the filters whenever the filters are evaluated.

▼ To Create a New Static Group

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Create a new static group using the `ldapmodify` command.**

   For example, to create a new static group called System Administrators and to add some members, you could use this command:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=System Administrators, ou=Groups, dc=example,dc=com
changetype: add
cn: System Administrators
objectclass: top
objectclass: groupOfNames
ou: Groups
member: uid=kvaughan, ou=People, dc=example,dc=com
member: uid=rdaugherty, ou=People, dc=example,dc=com
member: uid=hmiller, ou=People, dc=example,dc=com
```

2. **Check to see that the new group has been created and that the members have been added.**

   For example, to check that Kirsten Vaughan is in the new System Administrators group, type:

```
$ ldapsearch -b "dc=example,dc=com" uid=kvaughan isMemberOf
uid=kvaughan,ou=People,dc=example,dc=com
isMemberOf: cn=System Administrators, ou=Groups, dc=example,dc=com
isMemberOf: cn=HR Managers,ou=groups,dc=example,dc=com
```

▼ To Create a New Dynamic Group

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Create a new dynamic group by using the `ldapmodify` command.**

  For example, to create a new dynamic group called "3rd Floor", which includes all employees whose room numbers start with 3, you could use this command:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=3rd Floor, ou=Groups, dc=example,dc=com
changetype: add
cn: 3rd Floor
objectclass: top
objectclass: groupOfUrls
ou: Groups
memberURL: ldap:///dc=example,dc=com??sub?(roomnumber=3*)
```

Managing Roles

A role is an alternate grouping mechanism that is designed to be more efficient and easier for applications to use. While roles are defined and administered like groups, a generated role attribute in each member entry automatically indicates the roles of an entry. For example, an application can read the roles of an entry, rather than having to select a group and browse the members list.

By default, the scope of a role is limited to the subtree where the scope is defined. However, you can extend scoping of the nested role. You can allow the scope to nest roles located in other subtrees and to have members anywhere in the directory. For details see "To Extend the Scope of a Role" on page 239 and "Example of a Nested Role Definition" on page 238.

This section explains how to use roles securely, and how to manage roles from the command line.

Using Roles Securely

To use roles securely, you must set access control instructions (ACIs) to protect appropriate attributes. For example, user A possesses the managed role, MR. Managed roles are equivalent to static groups, and explicitly assign a role to each member entry by adding the `nsRoleDN` attribute to the entry. The MR role has been locked using account inactivation through the command line. This means that user A cannot bind to the server because the `nsAccountLock` attribute is computed as "true" for that user. However, suppose the user was already bound and noticed that he is now locked through the MR role. If no ACI exists to prevent the user from having write access to the `nsRoleDN` attribute, the user can remove the `nsRoleDN` attribute from his own entry and unlock himself.

To prevent users from removing the `nsRoleDN` attribute, you must apply ACIs. With filtered roles, you must protect the part of the filter that would prevent the user from being able to relinquish the filtered role by modifying an attribute. Users should not be allowed to add, delete, or modify the attribute used by the filtered role. In the same way, if the value of the filter attribute is computed, all the attributes that can modify the value of the filter attribute need to be protected. As nested roles can contain filtered and managed roles, the preceding points should be considered for each of the roles that are contained in the nested role.

For detailed instructions on setting ACIs for security, see Chapter 6, "Directory Server Access Control."

Managing Roles From the Command Line

Roles are defined in entries that the Directory Administrator can access through command-line utilities. After you create a role, you assign members to the role as follows:

- Members of a managed role have the `nsRoleDN` attribute in their entry.
- Members of a filtered role are entries that match the filter specified in the `nsRoleFilter` attribute.
- Members of a nested role are members of the roles that are specified in the `nsRoleDN` attributes of the nested role definition entry.

All role definitions inherit from the `LDAPsubentry` and `nsRoleDefinition` object classes. The following example shows additional object classes and associated attributes specific to each type of role.

Example of a Managed Role Definition

To create a role for all marketing staff, use the following `ldapmodify` command:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition
cn: Marketing
description: managed role for marketing staff
```

Notice that the `nsManagedRoleDefinition` object class inherits from the `LDAPsubentry`, `nsRoleDefinition`, and `nsSimpleRoleDefinition` object classes.

Assign the role to a marketing staff member who is named Bob by updating his entry as follows:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Bob Arnold,ou=marketing,ou=People,dc=example,dc=com
changetype: modify
add: nsRoleDN
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
```

The `nsRoleDN` attribute indicates that the entry is a member of a managed role. The managed role is identified by the DN of its role definition. To allow users to modify their own `nsRoleDN` attribute, but to prevent users from adding or removing the `nsManagedDisabledRole`, add the following ACI:

```
aci: (targetattr="nsRoleDN")(targattrfilters="add=nsRoleDN:
(!(nsRoleDN=cn=AdministratorRole,dc=example,dc=com)),
```

```
del=nsRoleDN:(!(nsRoleDN=cn=nsManagedDisabledRole,dc=example, dc=com)")
(version3.0;aci "allow mod of nsRoleDN by self except for critical values";
allow(write) userdn="ldap:///self";)
```

Example of a Filtered Role Definition

To set up a filtered role for sales managers, assuming that they all have the `isManager` attribute, use the following `ldapmodify` command:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerFilter
nsRoleFilter: (isManager=True)
Description: filtered role for sales managers
```

Notice that the `nsFilteredRoleDefinition` object class inherits from the `LDAPsubentry`, `nsRoleDefinition`, and `nsComplexRoleDefinition` object classes. The `nsRoleFilter` attribute specifies a filter that finds all employees in the `ou=sales` organization that have subordinates, for example:

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=sales,ou=People,dc=example,dc=comcn: Carla Fuentes
isManager: TRUE...
nsRole: cn=ManagerFilter,ou=sales,ou=People,
dc=example,dc=com
```

Note – The filter string of a filtered role can be based on any attribute, except computed attributes that are generated by the CoS mechanism.

When filtered role members are user entries, you can choose to restrict their ability to add or remove themselves from the role. Protect the filtered attributes with ACIs.

Example of a Nested Role Definition

The roles that are nested within the nested role are specified by using the `nsRoleDN` attribute. Use the following command to create a role that contains both the marketing staff and sales manager members of the roles created in the previous examples:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=MarketingSales,ou=marketing,ou=People,dc=example,dc=com
```

```
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
cn: MarketingSales
nsRoleDN: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
nsRoleScopeDN: ou=sales,ou=People,dc=example,dc=com
```

Notice that the `nsNestedRoleDefinition` object class inherits from the `LDAPsubentry`, `nsRoleDefinition`, and `nsComplexRoleDefinition` object classes. The `nsRoleDN` attributes contain the DN of the marketing managed role and the sales managers filtered role. Both of the users in the previous examples, Bob and Carla, would be members of this new nested role.

The scope of this filter includes the default scope, which is the subtree where the filter is located, and the subtree below any values of the `nsRoleScopeDN` attribute. In this case, the `ManagerFilter` is in the `ou=sales,ou=People,dc=example,dc=com` subtree. This subtree must be added to the scope.

Extending the Scope of a Role

Directory Server provides an attribute that allows the scope of a role to be extended beyond the subtree of the role definition entry. This single-valued attribute, `nsRoleScopeDN`, contains the DN of the scope to be added to an existing role. The `nsRoleScopeDN` attribute can only be added to a nested role. See "Example of a Nested Role Definition" on page 238.

▼ To Extend the Scope of a Role

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

The `nsRoleScopeDN` attribute enables you to extend the scope of a role in one subtree to include an entry in another subtree. For example, imagine two main subtrees in the example.com directory tree: `o=eng,dc=example,dc=com` (the engineering subtree) and `o=sales,dc=example,dc=com` (the sales subtree.) A user in the engineering subtree requires access to a sales application governed by a role in the sales subtree (`SalesAppManagedRole`). To extend the scope of the role, do the following:

1. **Create a role for the user in the engineering subtree.**

   For example, create the role `EngineerManagedRole`. This example uses a managed role, but it could just as well have been a filtered or nested role.

2. **Create a nested role, for example, `SalesAppPlusEngNestedRole`, in the sales subtree to house the newly created `EngineerManagedRole` and the initial `SalesAppManagedRole`.**

3 **Add the `nsRoleScopeDN` attribute to the `SalesAppPlusEngNestedRole`, with the DN of the engineering subtree scope that you want to add, in this case, `o=eng,dc=example,dc=com`.**

The necessary permissions must be granted to the engineering user so that he can access the `SalesAppPlusEngNestedRole` role and, in turn, the sales application. In addition, the entire scope of the role must be replicated.

Note – The restriction of extended scope to nested roles means that an administrator who previously managed roles in one domain only has rights to use the roles that already exist in the other domain. The administrator is not able to create an arbitrary role in the other domain.

Class of Service

The class of service (CoS) mechanism generates computed attributes as an entry is retrieved for a client application, which simplifies entry management and reduces storage requirements. The CoS mechanism allows attributes to be shared between entries, and as with groups and roles, CoS relies on helper entries.

For an explanation of how you can use CoS in your deployment, see Chapter 12, "Directory Server Class of Service," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

Note – Any search operation can test the existence of a CoS-generated attribute or compare the value of the attribute. The names of the computed attributes may be used in any filter string from a client search operation, except in an internal filter used in a filtered role.

You must restart the Directory Server instance to consider the new CoS definitions.

Using CoS Securely

The following sections describe the general principles for read and write protection of data in each of the CoS entries. The detailed procedure for defining individual access control instructions (ACIs) is described in Chapter 6, "Directory Server Access Control."

Protecting the CoS Definition Entry

Although the CoS definition entry does not contain the value of the generated attribute, it does provide the information to find that value. Reading the CoS definition entry reveals how to find the template entry that contains the value. Writing to this entry modifies how the computed attribute is generated.

You should therefore define both read and write ACIs for the CoS definition entries.

Protecting the CoS Template Entries

The CoS template entry contains the value of the generated CoS attribute. Therefore, at a minimum, the CoS attribute in the template must be protected by an ACI for both reading and updating.

- In the case of *pointer CoS*, the single template entry should not be allowed to be renamed. In most cases, it is simplest to protect the entire template entry.
- With *classic CoS*, all template entries have a common parent specified in the definition entry. If only templates are stored in this parent entry, access control to the parent entry protects the templates. However, if other entries beneath the parent require access, the template entries must be protected individually.
- In the case of *indirect CoS*, the template can be any entry in the directory, including user entries that might still need to be accessed. Depending on your needs, you can either control access to the CoS attribute throughout the directory or ensure that the CoS attribute is secure in each entry that is used as a template.

Protecting the Target Entries of a CoS

All entries in the scope of a CoS definition, for which the computed CoS attribute is generated, also contribute to computing its value.

When the CoS attribute already exists in a target entry, by default, the CoS mechanism does not override this value. If you do not want this behavior, define your CoS to override the target entry, or protect the CoS attribute in all potential target entries.

Both indirect and classic CoS also rely on a specifier attribute in the target entry. This attribute specifies the DN or RDN of the template entry to use. You should use an ACI to protect this attribute either globally throughout the scope of the CoS or individually on each target entry where it is needed.

Protecting Other Dependencies

Computed CoS attributes can be defined in terms of other generated CoS attributes and roles. You must understand and protect these dependencies to ensure that your computed CoS attribute is protected.

For example, the CoS specifier attribute in a target entry could be `nsRole`. Therefore the role definition must also be protected by an ACI.

In general, any attribute or entry that is involved in the computation of the computed attribute value should have an ACI for both read and write access control. For this reason, complex dependencies should be well planned or simplified to reduce subsequent complexity of access control implementation. Keeping dependencies on other computed attributes to a minimum improves directory performance and reduces maintenance.

Managing CoS From the Command Line

Because all configuration information and template data are stored as entries in the directory, you can use the LDAP command-line tools to configure and manage CoS definitions. This section shows how to create CoS definition entries and CoS template entries from the command line.

Creating the CoS Definition Entry From the Command Line

All CoS definition entries have the `LDAPsubentry` object class and inherit from the `cosSuperDefinition` object class. In addition, each type of CoS inherits from specific object classes and contains the corresponding attributes. The following table lists the object classes and attributes that are associated with each type of CoS definition entry.

TABLE 9–1 Object Classes and Attributes in CoS Definition Entries

CoS Type	CoS Definition Entry

Pointer CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosPointerDefinition`
	`cosTemplateDN:` *DN*
	`cosAttribute:` *attributeName override merge*

Indirect CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosIndirectDefinition`
	`cosIndirectSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

Classic CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosClassicDefinition`
	`cosTemplateDN:` *DN*
	`cosSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

In all cases, `cosAttribute` is multivalued. Each value defines an attribute that is generated by the CoS mechanism.

You can use the following attributes in CoS definition entries. For more information about each of these attributes, see the individual attributes in *Sun Directory Server Enterprise Edition 7.0 Man Page Reference*.

TABLE 9–2 CoS Definition Entry Attributes

Attribute	Purpose Within the CoS Definition Entry
`cosAttribute` *attributeName override merge*	Defines the name of the computed attribute for which you want to generate a value. This attribute is multivalued, and each value represents the name of an attribute whose value is generated from the template. The *override* and *merge* qualifiers specify how the CoS attribute value is computed in special cases described following this table. The *attributeName* cannot contain any subtypes. Attribute names with subtypes are ignored, but other values of `cosAttribute` are processed.
`cosIndirectSpecifier` *attributeName*	Defines the name of the attribute in target entries whose value is used by indirect CoS to identify the template entry. The named attribute is called the specifier and must contain a full DN string in each target entry. This attribute is single-valued, but the *attributeName* can be multivalued to designate multiple templates.
`cosSpecifier` *attributeName*	Defines the name of the attribute in target entries whose value is used by classic CoS to identify the template entry. The named attribute is called the specifier and must contain a string that can be found in the RDN of template entries. This attribute is single-valued, but the *attributeName* can be multivalued to designate multiple templates.
`cosTemplateDN` *DN*	Provides the full DN of the template entry for a pointer CoS definition or the base DN of the template entry for classic CoS. This attribute is single-valued.

Note – You cannot use the `isMemberOf` attribute as a `CosSpecifier` to make all the members of static groups automatically inherit from a common computed attribute value.

The `cosAttribute` attribute allows two qualifiers following the name of the CoS attribute, the override qualifier and the merge qualifier.

The *override* qualifier describes the behavior when an attribute that is dynamically generated by CoS already physically exists in the entry. The *override* qualifier can be one of the following:

- `default` (or no qualifier) - Indicates that the server does not override a real attribute value stored in the entry when the attribute is of the same type as the computed attribute.
- `override` - Indicates that the server always returns the value generated by the CoS, even when a value is stored with the entry.

- `operational` - Indicates that the attribute will only be returned if it is explicitly requested in the search. Operational attributes do not need to pass a schema check to be returned. The `operational` qualifier has the same behavior as the `override` qualifier.

  You can only make an attribute operational if the attribute is also defined as operational in the schema. For example, if your CoS generates a value for the `description` attribute, you cannot use the `operational` qualifier because the `description` attribute is not marked operational in the schema.

The *merge* qualifier is either absent or `merge-schemes`. This qualifier allows the computed CoS attribute to be multivalued, either from multiple templates or multiple CoS definitions. For more information, see "Multivalued CoS Attributes" on page 244.

Overriding Real Attribute Values

You might create a pointer CoS definition entry that contains an `override` qualifier as follows:

```
dn: cn=pointerCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=exampleUS,cn=data
cosAttribute: postalCode override
```

This pointer CoS definition entry indicates that the entry is associated with the template entry `cn=exampleUS,cn=data` that generates the value of the `postalCode` attribute. The `override` qualifier indicates that this value takes precedence over the value of the `postalCode` attribute if the attribute exists in a target entry.

Note – If the CoS attribute is defined with the `operational` or `override` qualifiers, you cannot perform write operations on the "real" value of that attribute in any entry in the CoS scope.

Multivalued CoS Attributes

When you specify the `merge-schemes` qualifier, the generated CoS attribute can be multivalued in two ways:

- With indirect or classic CoS, the specifier attributes in target entries can be multivalued. In this case, each value determines a template, and the value from each template is part of the generated value.
- Multiple CoS definition entries of any type can contain the same attribute name in their `cosAttribute`. In this case, if all definitions contain the `merge-schemes` qualifier, the generated attribute contains all values computed by each definition.

The two situations can occur together and define even more values. However, in all cases, duplicate values will only be returned one time in a generated attribute.

In the absence of the `merge-schemes` qualifier, the `cosPriority` attribute of the template entry is used to determine a single value among all templates for the generated attribute. This scenario is described in the next section.

The `merge-schemes` qualifier never merges a "real" value that is defined in the target with generated values from the templates. The *merge* qualifier is independent of the *override* qualifier. All pairings are possible, and the behaviors implied by each are complimentary. Also, the qualifiers can be specified in any order after the attribute name.

Note – When there are multiple CoS definitions for the same attribute, the definitions must all have the same *override* and *merge* qualifiers. When different pairs of qualifiers occur in CoS definitions, one of the combinations is selected arbitrarily among all definitions.

CoS Attribute Priority

If multiple CoS definitions or multivalued specifiers exist, but no `merge-schemes` qualifier, Directory Server uses a priority attribute to select a single template that defines the single value of the computed attribute.

The `cosPriority` attribute represents the global priority of a particular template among all those being considered. A priority of zero is the highest priority. Templates that contain no `cosPriority` attribute are considered the lowest priority. When two or more templates provide an attribute value but have the same or no priority, a value is chosen arbitrarily.

Template priorities are not taken into account when using the `merge-schemes` qualifier. When merging, all templates being considered define a value regardless of any priority that the templates define. The `cosPriority` attribute is defined on CoS template entries as described in the following section.

Note – The `cosPriority` attribute must not have a negative value. Also, attributes generated by indirect CoS do not support priority. Do not use `cosPriority` in template entries of an indirect CoS definition.

Creating the CoS Template Entry From the Command Line

When using pointer CoS or classic CoS, the template entry contains the `LDAPsubentry` and `cosTemplate` object classes. This entry must be created specifically for the CoS definition. Making the CoS template entry an instance of the `LDAPsubentry` object class allows ordinary searches to be performed unhindered by the configuration entries.

The template of the indirect CoS mechanism is an arbitrary, existing entry in the directory. The target does not need to be identified ahead of time or given the `LDAPsubentry` object class, but the target must have the auxiliary `cosTemplate` object class. The indirect CoS template is accessed only when the CoS is evaluated to generate a computed attribute and its value.

In all cases, the CoS template entry must contain the attribute and the value that is generated by the CoS on the target entries. The attribute name is specified in the `cosAttribute` attribute of the CoS definition entry.

The following example shows a template entry of the highest priority for a pointer CoS that generates the `postalCode` attribute:

```
dn: cn=ZipTemplate,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 95054
cosPriority: 0
```

The following sections provide examples of template entries along with examples of each type of CoS definition entry.

Example of a Pointer CoS

The following command creates a pointer CoS definition entry that has the `cosPointerDefinition` object class. This definition entry uses the CoS template entry that is stated in the example in the previous section to share a common postal code among all entries in the `ou=People,dc=example,dc=com` tree.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=pointerCoS,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=ZipTemplate,ou=People,dc=example,dc=com
cosAttribute: postalCode
```

The CoS template entry (`cn=ZipTemplate,ou=People,dc=example,dc=com`) supplies the value stored in its `postalCode` attribute to all entries located under the `ou=People,dc=example,dc=com` suffix. If you search for any entry that does not have a postal code in the same subtree, you will see the value of the generated attribute:

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
postalCode: 95054
```

Example of an Indirect CoS

Indirect CoS names an attribute in the `cosIndirectSpecifier` attribute to locate the template specific to each target. The template entry for indirect CoS may be any entry in the directory, including other user entries. This example indirect CoS uses the `manager` attribute of the target entry to identify the CoS template entry. The template entry is the manager's user entry. The manager's user entry contains the value of the attribute to generate. The value is that of `departmentNumber` in this case.

The following command creates the indirect CoS definition entry, which contains the `cosIndirectDefinition` object class:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=generateDeptNum,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosIndirectDefinition
cosIndirectSpecifier: manager
cosAttribute: departmentNumber
```

Next, add the `cosTemplate` object class to the template entries, and make sure that they define the attribute to be generated. In this example, all manager entries are templates:

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Carla Fuentes,ou=People,dc=example,dc=com
changetype: modify
add: objectclass
objectclass: cosTemplate
-
add: departmentNumber
departmentNumber: 318842
```

With this CoS, target entries (the entries under `ou=People,dc=example,dc=com`) that contain the `manager` attribute automatically have the department number of their manager. The `departmentNumber` attribute is computed on the target entries because it does not exist in the server. However, the `departmentNumber` attribute is returned as part of the target entry. For example, if Babs Jensen's manager is defined to be Carla Fuentes, her department number is the following:

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
manager: cn=Carla Fuentes,ou=People,dc=example,dc=com
departmentNumber: 318842
```

Example of a Classic CoS

This example shows how to generate a postal address with a classic CoS. The generated value is specified in a template entry that is located by a combination of the `cosTemplateDN` in the CoS definition and the value of the `cosSpecifier` attribute in the target entry. The following command creates the definition entry by using the `cosClassicDefinition` object class:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=classicCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: ou=People,dc=example,dc=com
cosSpecifier: building
cosAttribute: postalAddress
```

Using the same command, create the template entries that give the postal address for each building:

```
dn: cn=B07,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalAddres: 7 Old Oak Street, Anytown, CA 95054
```

With this CoS, target entries (the entries under `ou=People,dc=example,dc=com`) that contain the `building` attribute will automatically have the corresponding postal address. The CoS mechanism searches for a template entry that has the specifier attribute value in its RDN. In this example, if Babs Jensen is assigned to building B07, her postal address is generated as follows:

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
building: B07
postalAddress: 7 Old Oak Street, Anytown, CA 95054
```

Creating Role-Based Attributes

You can create classic CoS schemes that generate attribute values for an entry that is based on the role possessed by the entry. For example, you could use role-based attributes to set the server look-through limit on an entry-by-entry basis.

To create a role-based attribute, use the `nsRole` attribute as the `cosSpecifier` in the CoS definition entry of a classic CoS. Because the `nsRole` attribute can be multivalued, you can define CoS schemes that have more than one possible template entry. To resolve the ambiguity of which template entry to use, you can include the `cosPriority` attribute in your CoS template entry.

For example, you can create a CoS that allows members of the manager role to exceed the standard mailbox quota. The manager role is as follows:

```
dn: cn=ManagerRole,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerRole
nsRoleFilter: (isManager=True)
Description: filtered role for managers
```

The classic CoS definition entry is created as follows:

```
dn: cn=generateManagerQuota,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: cn=managerCOS,ou=People,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: mailboxquota override
```

The CoS template name must be a combination of the `cosTemplateDn` and the value of `nsRole`, which is the DN of the role. For example:

```
dn:cn="cn=ManagerRole,ou=People,dc=example,dc=com",\
 cn=managerCOS,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
mailboxquota: 1000000
```

The CoS template entry provides the value for the `mailboxquota` attribute. An additional qualifier of `override` tells the CoS to override any existing `mailboxquota` attributes values in the target entry. Target entries that are members of the role will have computed attributes generated by the role and by the CoS, for example:

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -\
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
```

```
dn: cn=Carla Fuentes,ou=People,dc=example,dc=comcn: Carla Fuentes
isManager: TRUE...nsRole: cn=ManagerRole,ou=People,dc=example,dc=com
mailboxquota: 1000000
```

Note – The role entry and the CoS definition entry should be located in the same place in the directory tree so that they have the same target entries in their scope. The CoS target entry should also be located in the same place so that it is easy to find and maintain.

Monitoring the CoS Plug-In

Directory Server enables you to monitor certain aspects of the CoS plug-in. CoS monitoring attributes are stored in the `cn=monitor,cn=Class of Service,cn=plugins,cn=config` entry. For details of the each attribute under this entry and the information that they provide, see *Sun Directory Server Enterprise Edition 7.0 Man Page Reference*.

Setting CoS Logging

Directory Server logs warning messages when it is forced to make an arbitrary distinction among multiple applicable definition entries. Such warning messages takes this form:

```
Definition /defDN1/ and definition /defDN2/ compete to provide attribute
 '/type/' at priority /level/
```

You can also configure Directory Server to log informational messages when the server is forced to make an arbitrary distinction among multiple, potentially applicable definition entries. To do so, set the error log to include messages from plug-ins.

Note – Because setting additional log levels can result in a heavy logging load, you might not want to set logging on a production server.

The content of informational messages takes the following form:

```
Definition /defDN1/ and definition /defDN2/ potentially compete
to provide attribute '/type/' at priority /level/
```

You can then choose whether to resolve such cases of CoS ambiguity by setting CoS priorities appropriately on the definition entries.

Maintaining Referential Integrity

Referential integrity is a plug-in mechanism that ensures that relationships between entries are maintained. Several types of attributes, such as those for group membership, contain the DN of another entry. Referential integrity can be used to ensure that when an entry is removed, all attributes that contain its DN are also removed.

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 must be manually removed from the group by the administrator. This is an important feature if you are integratingDirectory Server with other Sun products that rely on the directory for user and group management.

How Referential Integrity Works

When the referential integrity plug-in is enabled it performs integrity updates on specified attributes immediately after a delete, rename, or move operation. By default, the referential integrity plug-in is disabled.

Whenever you delete, rename, or move a user or group entry in the directory, the operation is logged to the referential integrity log file:

instance-path`/logs/referint`

After a specified time, known as the *update interval*, the server performs a search on all attributes for which referential integrity is enabled, and matches the entries resulting from that search with the DNs of deleted or modified entries present in the log file. 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.

When the default configuration of the referential integrity plug-in is enabled, it performs integrity updates on the `member`, `uniquemember`, `owner`, `seeAlso`, and `nsroledn` attributes immediately after a delete, rename, or move operation. You can, however, configure the behavior of the referential integrity plug-in to suit your own requirements. The following behavior can be configured:

- Record referential integrity updates in a different file.
- Modify the update interval.

  If you want to reduce the impact that referential integrity updates has on your system, you might want to increase the amount of time between updates.
- Select the attributes to which you apply referential integrity.

  If you use or define attributes containing DN values, you might want the referential integrity plug-in to monitor them.

▼ To Configure the Referential Integrity Plug-In

Note – All attributes in all databases that are used by the referential integrity plug-in must be indexed. The indexes need to be created in the configuration of all the databases. When the retro changelog is enabled, the `cn=changelog` suffix must be indexed. For information, see Chapter 12, "Directory Server Indexing."

Certain limitations are associated with using the referential integrity plug-in in a replicated environment. For a list of these limitations, see "Replication and Referential Integrity" on page 275.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Make sure that all replicas are configured and that all replication agreements are defined.**

2. **Determine the set of attributes for which you will maintain referential integrity and the update interval that you want to use on your master servers.**

3. **Enable the referential integrity plug-in on all master servers using the same set of attributes and the same update interval.**

   - To define the attributes for referential integrity, use this command:

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr:attribute-name \
      ref-integrity-attr:attribute-name
     ```

   - To add a referential integrity attribute to an existing list of attributes, use this command:

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr+:attribute-name
     ```

   - To define the referential integrity update interval, use this command:

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-check-delay:duration
     ```

   - To enable referential integrity, use this command:

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-enabled:on
     ```

4. **Ensure that the referential integrity plug-in is disabled on all consumer servers.**

◆ ◆ ◆ CHAPTER 10

Directory Server Replication

Replication is the mechanism by which directory contents are automatically copied from a Directory Server to one or more other Directory Servers. All write operations are automatically mirrored to other Directory Servers. Replication between different supported platforms is possible. For a list of supported platforms, see "Platform Support" in *Sun Directory Server Enterprise Edition 7.0 Release Notes*. For a complete description of replication concepts, replication scenarios, and how to plan for replication in your directory deployment, see the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

In a replication topology, generally one suffix on a server is replicated to or from another suffix on a server. For this reason, the terms replica, replicated suffix and replicated server can be used interchangeably.

This chapter describes the tasks to be performed to set up the various replication scenarios by using the command line, and covers the following topics:

Planning Your Replication Deployment

You can configure a replication deployment with an unlimited number of masters. You are not required to include hubs or consumers in your deployment. Procedures for configuring replication for hubs and consumers are included in this chapter, but they are optional.

Before you begin configuring replication, you need to have a clear understanding of the way that replication will be deployed in your organization. You must understand the replication concepts described in *Sun Directory Server Enterprise Edition 7.0 Reference*. You must also have carefully planned your future replication configuration using the design guidelines provided in the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

Recommended Interface for Configuring and Managing Replication

The easiest way to configure and manage replication is by using Directory Service Control Center (DSCC). By using DSCC, you can configure replication automatically. You can choose the level of automation you require for setting up your replication topology, for example, whether you want to initialize the suffixes during replication configuration or not. DSCC also provides checks that can prevent errors. In addition, DSCC provides a graphical view of the replication topology.

The DSCC online help provides procedures for setting up replication by using DSCC.

Note – Only use the command-line procedures provided in this chapter if you are unable to use DSCC for configuring replication.

Summary of Steps for Configuring Replication

"Summary of Steps for Configuring Replication" on page 255 assumes that you are replicating a single suffix. If you are replicating more than one suffix, you can configure the suffixes in parallel on each server. In other words, you can repeat each step to configure replication on multiple suffixes.

The rest of this chapter contains detailed instructions on how to configure replication.

▼ Summary of Steps for Configuring Replication

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

To configure any replication topology, follow the general steps as outline in this procedure.

1 **Do the following on all servers that contain a dedicated consumer replica:**

   a. **Create an empty suffix for the consumer replicated suffix.**
      See "To Create a Suffix for a Consumer Replica" on page 257.

   b. **Enable the consumer replicated suffix.**
      See "To Enable a Consumer Replica" on page 257.

   c. **(Optional) Configure the advanced consumer settings.**
      See "To Perform Advanced Consumer Configuration" on page 257.

2 **If applicable, do the following on all servers that contain a hub replicated suffix:**

   a. **Create an empty suffix for the hub replicated suffix.**
      See "To Create a Suffix for a Hub Replica" on page 259.

   b. **Enable the hub replicated suffix.**
      See "To Enable a Hub Replica" on page 259.

   c. **(Optional) Configure the advanced hub settings.**
      See "To Modify Change Log Settings on a Hub Replica" on page 259.

3 **Do the following on all servers that contain a master replicated suffix:**

   a. **Create a suffix for the master replicated suffix.**
      See "To Create a Suffix for a Master Replica" on page 260.

   b. **Enable the master replicated suffix.**
      See "To Enable a Master Replica" on page 260.

   c. **(Optional) Configure the advanced master settings.**
      See "To Modify Change Log Settings on a Master Replica" on page 260.

Note – Make sure that you enable all replicas before you create a replication agreement so that you can initialize consumer replicas immediately after you create the replication agreement. Consumer initialization is always the last stage in setting up replication.

4 **Ensure your replication manager configuration is complete.**

- If you plan to use the default manager, set the default replication manager password on all servers. See "To Change the Default Replication Manager Password" on page 263.
- If you plan to use a non-default replication manager, define the alternative replication manager entry on all servers. See "Using a Non-Default Replication Manager" on page 261.

5 **Create replication agreements on all master replicas as follows:**

a. **Between masters in a multimaster topology**

b. **Between masters and their dedicated consumers**

c. **Between masters and hub replicas**

See "Creating and Changing Replication Agreements" on page 263.

6 **(Optional) If you want to use fractional replication, configure it now.**

See "Fractional Replication" on page 265.

7 **(Optional) If you want to use replication priority, configure it now.**

See "Replication Priority" on page 266.

8 **Configure replication agreements between the hub replicas and their consumers.**

See "Creating and Changing Replication Agreements" on page 263.

9 **For multimaster replication, initialize all masters from the same master replica that contains the original copy of the data.**

See "Initializing Replicas" on page 267.

10 **Initialize the hub and consumer replicas.**

See "Initializing Replicas" on page 267.

Enabling Replication on a Dedicated Consumer

A dedicated consumer is a read-only copy of a replicated suffix. The dedicated consumer receives updates from servers that bind as the replication manager to make changes. Configuring the consumer server consists of preparing an empty suffix to hold the replicated suffix and enabling replication on that suffix. Optional advanced configuration can include setting referrals, changing the purge delay, and modifying properties.

The following sections explain how to configure one dedicated consumer replicated suffix on its server. Repeat all procedures on each server that will contain a dedicated consumer replicated suffix.

▼ To Create a Suffix for a Consumer Replica

- **If an empty suffix does not already exist, create it on the consumer with the same DN as the intended master replica.**

  For instructions, see "Creating Suffixes" on page 47.

Caution – If the suffix exists and is not empty, its contents will be lost when the replicated suffix is initialized from the master.

▼ To Enable a Consumer Replica

After you have created an empty suffix, you need to enable the consumer replicated suffix.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Enable the consumer replicated suffix.**

  ```
  $ dsconf enable-repl -h host -p port consumer suffix-DN
  ```

  For example:

  ```
  $ dsconf enable-repl -h host1 -p 1389 consumer dc=example,dc=com
  ```

▼ To Perform Advanced Consumer Configuration

If you want to configure your consumer replicated suffix for advanced features, do so now.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **If you want to use SSL for referrals, set secure referrals.**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:ldaps://servername:port
   ```

   For example:

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
    referral-url:ldaps://server2:2389
   ```

   The replication mechanism automatically configures consumers to return referrals for all known masters in the replication topology. These default referrals assume that clients will use simple authentication over a regular connection. If you want to give clients the option of binding to masters using SSL for a secure connection, add referrals of the form `ldaps://`*servername* `:`*port* that use a secure *port* number. Note that if the masters are configured for secure connections only, the URLs will point to the secure ports by default.

   If you have added one or more LDAP URLs as referrals, you can force the consumer to send referrals exclusively for these LDAP URLs and not for the master replicas. For example, suppose that you want clients to always be referred to the secure port on the master servers and not to the default port. Create a list of LDAP URLs for these secure ports, and set the property for using these referrals. You can also use an exclusive referral if you want to designate a specific master or a Directory Server proxy to handle all updates.

2. **If you want to change the replication purge delay for the consumer, use this command:**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN repl-purge-delay:time
   ```

   For example, to set the purge delay to 2 days, type:

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 edc=example,dc=com repl-purge-delay:2d
   ```

Enabling Replication on a Hub

Hub replicas act as both consumers and masters to further distribute replicated data to a larger number of consumers. Hub replicas receive replication updates from their suppliers and send replication updates to their consumers. They do not accept modifications, but instead return referrals to the masters.

Configuring a hub server consists of preparing an empty suffix to hold the replicated suffix and enabling replication on that suffix. Optional advanced configuration can include choosing a different replication manager, setting referrals, setting the purge delay, and modifying change log parameters.

The following sections explain how to configure one hub server. Repeat all procedures on each server that will contain a hub replicated suffix.

To Create a Suffix for a Hub Replica

- **If an empty suffix does not already exist, create it on the hub server with the same DN as the intended master replica.**

  For instructions, see "Creating Suffixes" on page 47.

  If the suffix exists and is not empty, its contents will be lost when the replicated suffix is initialized from the master.

To Enable a Hub Replica

If you have hub replicas, enable them now.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Enable the hub replicated suffix.**

  ```
  $ dsconf enable-repl -h host -p port hub suffix-DN
  ```

  For example:

  ```
  $ dsconf enable-repl -h host1 -p 1389 hub dc=example,dc=com
  ```

To Modify Change Log Settings on a Hub Replica

For advanced hub configuration, the only parameters that you might want to modify are related to the change log. As a supplier, a hub server requires a change log.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **To modify a change log setting on a hub, use one of the following commands:**

  ```
  $ dsconf set-suffix-prop -h host -p port suffix-DN repl-cl-max-age:value

  $ dsconf set-suffix-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
  ```

Enabling Replication on a Master Replica

Master replicas contain the master copy of the data and centralize all modifications before propagating updates to all other replicas. A master records all changes, checks the status of its consumers, and sends updates to them when necessary. In multimaster replication, master replicas also receive updates from other masters.

Configuring a master server consists of defining the suffix that contains the master replica, enabling the master replica, and configuring it for advanced replication, if necessary.

The following sections explain how to configure one master server. Repeat all procedures on each server that will contain a master replicated suffix.

▼ To Create a Suffix for a Master Replica

- **Choose or create a suffix on the master server that will contain the entries that you want to replicate.**

  For instructions, see to "Creating Suffixes" on page 47.

  To ensure correct multimaster configuration and initialization, only load one of the masters with the data. Any data on other replicated suffixes will be overwritten.

▼ To Enable a Master Replica

When you enable replication on a master, you must assign a replication ID. The replication ID is used to distinguish the owner of update statements and to resolve conflicts that might occur with multimaster replication. Therefore, the replication ID must be unique for all master replicas of this suffix. Once set, the replication ID must not be changed.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Enable the master replicated suffix.**

  ```
  $ dsconf enable-repl -h host -p port -d ReplicaID master suffix-DN
  ```

  where *ReplicaID* is an integer between 1 and 65534.

  For example, to create a master replicated suffix with replica ID 1, use this command:

  ```
  $ dsconf enable-repl -h host1 -p 1389 -d 1 master dc=example,dc=com
  ```

▼ To Modify Change Log Settings on a Master Replica

For advanced master configuration, you might want to modify the change log settings.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **If you want to modify a change log setting on a master, use one of the following commands:**

```
$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value

$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
```

Configuring the Replication Manager

This section describes how to configure a non-default replication manager and how to set the default replication manager password.

Using a Non-Default Replication Manager

The *replication manager* is the user that suppliers will use to bind to a consumer server when sending replication updates. All servers that contain suffixes receiving updates must have at least one replication manager entry.

Directory Server has a default replication manager entry that you can use on every server, especially for simple replication scenarios: `cn=replication manager,cn=replication,cn=config`. The replication mechanism automatically configures consumer replicas with this user, simplifying the deployment of replicas.

If you have a more complex replication scenario, you might want several replication managers with a different password for each replicated suffix. You can replace the existing default replication manager with one or more new replication managers.

Caution – Never bind or perform operations on the server using the DN and password of the replication manager. The replication manager is for use only by the replication mechanism. Any other use might require reinitializing the replicas.

Never use the Directory Manager as the replication manager. Because the cn=admin,cn=Administrators,cn=config entry is used for other administrative tasks, you must also not use this user or any other user in the administrator group as the replication manager.

After you have chosen the replication manager for each consumer, ensure that you remember the replication manager DN that you chose or created. You will need this DN and its password later when creating the replication agreement with this consumer on its supplier.

▼ To Set A Non-Default Replication Manager

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 On all consumer (destination) replicated suffixes, create a new replication manager and password.

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=new-replication-manager,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:password
sn:new-replication-manager
```

For example:

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=ReplicationManager3,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:secret
sn:ReplicationManager3
```

2 On all consumer (destination) replicated suffixes, set the replication manager bind DN.

```
$ dsconf set-suffix-prop -h host -p port suffix-DN \
 repl-manager-bind-dn:"cn=new-replication-manager,cn=replication,cn=config"
```

For example:

```
$ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
 repl-manager-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config"
```

3 For all replication agreements that you have created on all supplier (source) replicated suffixes, set the replication manager bind DN.

a. Create a temporary file for setting the new replication manager password.

This file is read once, and the password is stored for future use.

```
$ echo password > password-file
```

b. Set the replication manager bind DN and password to be used by the replication mechanism when performing updates.

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN host:port \
 auth-bind-dn:"cn=new-replication-manager,cn=replication,cn=config" \
 auth-pwd-file:password-file
```

For example:

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 auth-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config" \
 auth-pwd-file:pwd.txt
```

c. Remove the temporary password file.

```
$ rm password-file
```

▼ To Change the Default Replication Manager Password

1 Create a temporary file for setting the replication manager password.

This file is read once, and the password is stored for future use.

```
$ echo password > password-file
```

2 On all consumer (destination) servers in the replication topology, set the replication manager bind password.

```
$ dsconf set-server-prop -h host -p port def-repl-manager-pwd-file:password-file
```

For example:

```
$ dsconf set-server-prop -h host1 -p 1389 def-repl-manager-pwd-file:pwd.txt
```

3 Remove the temporary password file.

```
$ rm password-file
```

Creating and Changing Replication Agreements

A *replication agreement* is a set of parameters on a supplier that configures and controls how updates are sent to a given consumer. The replication agreement must be created on the supplier replicated suffix that is sending updates to its consumer. You must create a replication agreement on the supplier for every consumer that you want updated.

▼ To Create a Replication Agreement

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

If you use DSCC to create a new replication agreement, you can choose to copy some or all replication agreement configuration settings from an existing replication agreement.

1. **From your master server, create a replication agreement for each consumer that you want to replicate to.**

   ```
   $ dsconf create-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port\
    [consumer-host:consumer-port]
   ```

   For example:

   ```
   $ dsconf create-repl-agmt -h host1 -p 1389 dc=example,dc=com host2:1389
   ```

   To list existing replication agreements by using the command line, use the `dsconf list-repl-agmts` command.

   Note – If you change the port number on a master when replication is running, you do not need to reinitialize the servers. However, the old replication agreement that pointed to the old address (*host*:*old-port*) is no longer useful. If you want replication to continue as it did before the port number was changed, you must create a new agreement with the new address (*host*:*new-port*).

   ---

2. **Check that the replication agreement has been created correctly.**

   ```
   $ dsconf show-repl-agmt-status -h host -p port suffix-DN consumer-host:consumer-port
   ```

3. **If the authentication status is not OK, run the `dsconf accord-repl-agmt` command.**

   Note – Only use the command `dsconf accord-repl-agmt` if you are using the default replication manager. If you have created a new replication manager, do not use this command because it overwrites some required settings.

   The `dsconf accord-repl-agmt` command ensures that both the supplier and destination servers share the same replication authentication settings.

   ```
   $ dsconf accord-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
   ```

   For example:

   ```
   $ dsconf accord-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
   ```

▼ To Change the Destination of a Replication Agreement

This procedure changes the remote replica pointed to by an existing replication agreement. The suffix DN and configuration of the existing agreement remain the same.

- **Change the host name and port number of the remote replica in the replication agreement.**

  ```
  $ dsconf change-repl-dest -h host -p port suffix-DN host:port new-host:new-port
  ```

If this command is run with the `-A` *protocol* option, you can change the authentication protocol that is used by replication.

Fractional Replication

By default, the replication operation copies entire entries in the replicated suffix to consumer replicas. With the fractional replication feature, you can select the suffix that you want to use, and which attributes you want to include or exclude. Fractional replication is configured in the replication agreement, allowing you to define the attribute set for each consumer replicated suffix of a master. You can control which data is distributed and use replication bandwidth and consumer resources more efficiently.

For example, if you want to reduce replication bandwidth, you can choose not to replicate attributes with typically large values such as `photo`, `jpegPhoto`, and `audio`. As a result, these attributes will not be available on consumers. As another example, you can choose to replicate only the `uid` and `userpassword` attributes to a consumer server that is dedicated to performing authentication.

Considerations for Fractional Replication

Enabling or modifying a fractional set of attributes requires you to reinitialize the consumer replica. Therefore, you need to determine your fractional replication needs before deployment and define your attribute set before you initialize your replicated suffixes for the first time.

You need to proceed with caution when replicating a small set of attributes, given the dependency of complex features such as ACIs, roles, and CoS on certain attributes. In addition, not replicating other attributes that are mentioned in specifiers or filters of the ACI, roles, or CoS mechanisms might compromise the security of the data. Not replicating might also result in different sets of attributes being returned in searches. Managing a list of attributes to exclude is safer, and less prone to human error, than managing a list of attributes to include.

You need to turn off schema checking in the consumer server if the attribute set that you replicate does not allow all replicated entries to follow the schema. Replication of non-conforming entries does not cause errors because the replication mechanism bypasses schema checking on the consumer. However, the consumer will contain non-conforming entries and should have schema checking turned off to expose a coherent state to its clients.

Fractional replication is configured in the replication agreement of master replicas with hubs and dedicated consumers. Configuration of fractional replication between two master replicas in a multimaster replication environment is not supported. Also, if several masters have replication agreements with the same replica, all these agreements must replicate the same set of attributes.

▼ To Configure Fractional Replication

To configure fractional replication, you must specify the suffix, determine whether to include or exclude attributes on that suffix, then choose which attributes to include or exclude. If you choose to exclude attributes on a suffix, all other attributes are automatically included. Likewise, if you choose to include certain attributes on a suffix, all other attributes are automatically excluded.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure fractional replication on a replication agreement located on the source server.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port\
 property:value
```

where *property* is either `repl-fractional-exclude-attr` or `repl-fractional-include-attr`.

For example, if you want to configure a fractional agreement to exclude JPEG and TIFF photos from being replicated on the suffix `dc=example,dc=com`, use this command:

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389
 repl-fractional-exclude-attr:jpegPhoto repl-fractional-exclude-attr:tiffPhoto
```

To add an attribute to an existing list of attributes that should be excluded, use this command:

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port\
 repl-fractional-exclude-attr+:attribute
```

Replication Priority

Specifying replication priority is optional. You can create replication rules to specify that certain changes, such as updating the user password, are replicated with high priority. Any changes specified in replication rules are replicated as high priority, and all other changes are replicated with normal priority.

Note – Starting from version 6, replication priority rules only need to be created on the master server. No configuration is required for hubs and consumers.

▼ To Configure Replication Priority

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **To create a new replication priority rule on a master, use this command:**

  ```
  $ dsconf create-repl-priority -h host -p port suffix-DN priority-name property:value
  ```

  You can set replication priority with one or more of the following properties:

  - Operation type , `op-type`
  - Bind DN, `bind-dn`
  - Base DN, `base-dn`
  - Attribute type, `attr`

  The *priority-name* is user defined.

  For example, to create a replication rule specifying that user password changes are replicated with high priority, use this command:

  ```
  $ dsconf create-repl-priority -h host2 -p 1389 dc=example,dc=com pw-rule \
   attr:userPassword
  ```

  To display current replication rules, use the `dsconf list-repl-priorities -v` command. When used with the `-v` option, this command displays additional information related to prioritized replication rules.

  ```
  $ dsconf list-repl-priorities -h host2 -p 1389 -v
  ```

  For more information, see the `dsconf(1M)` man page.

Initializing Replicas

After you have created a replication agreement and after both replicas have been configured, you must initialize the consumer replicated suffix before replication can begin. During initialization, you physically copy data from the supplier replicated suffix to the consumer replicated suffix.

In addition, certain error conditions or configuration changes require you to reinitialize replicas. For example, if the data in a single master replicated suffix is restored from a backup for any reason, you need to reinitialize all of the replicas that it updates.

When reinitializing, the contents of the replicated suffix are deleted on the consumer and replaced with the contents of the suffix on the master. This ensures that the replicas are synchronized and that replication updates can resume. All of the initialization methods described in this section automatically rebuild the indexes of the consumer replica so that the consumer is ready to respond optimally to client read requests.

With multimaster replication, consumers might not need to be reinitialized if they have been updated by the other masters in the topology.

▼ To Initialize a Replicated Suffix from a Remote (Supplier) Server

You can initialize a suffix from a remote server by using an existing replication agreement. Use this method of initializing if possible, because it is less complicated than the other methods. Use the other methods only for large quantities of data that make the import too time consuming.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Online initialization of a replicated suffix by using DSCC is an easy way to initialize or reinitialize a consumer. However, if you are initializing a large number of entries, this process can be time consuming. In this case, you might find offline consumer initialization with the command line more efficient.

1. **Initialize your replica.**

   ```
   $ dsconf init-repl-dest -h host -p port suffix-DN destination-host:destination-port\
    [destination-host:destination-port]
   ```

   where *destination-host*:*destination-port* is the host and port of the destination server that you are initializing from the remote server.

2. **(Optional) For each agreement, check that the suffix appears as initialized.**

   ```
   $ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
   ```

Replica Initialization From LDIF

▼ To Initialize a Replicated Suffix From LDIF

This procedure outlines the general steps to use to initialize a replicated suffix from an LDIF file.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Online initialization of a replicated suffix by using DSCC is an easy way to initialize or reinitialize a consumer. However, if you are initializing a large number of entries, this process can be time consuming. In this case, you might find offline consumer initialization with the command line more efficient.

1. **Ensure that you have set up replication agreements.**

   You must do this *before* you initialize replicas.

2. **Export the original copy of the suffix data from a master replicated suffix to an LDIF file.**

   See "To Export a Replicated Suffix to LDIF" on page 269.

In a multimaster replication environment, you can use the LDIF file exported from the original master to initialize both the other masters and any consumers. In a cascading replication environment, you can use the same file to initialize both the hub replicas and their consumers.

In all cases, you must start with an LDIF file that has been exported from a configured master replica. You cannot use an arbitrary LDIF file to initialize all replicas because it does not contain replication metadata.

3 **If you are initializing a fractional replica, filter the file to keep only the replicated attributes, then transfer that file to all of the consumer servers.**

See "Filtering an LDIF File for Fractional Replication" on page 270.

4 **Initialize your replica.**

Do one of the following:

- For fast initialization on a server that is offline (stopped), use the `dsadm import` command.

  ```
  $ dsadm import instance-path LDIF_file suffix-DN
  ```

- To initialize a replica online from an LDIF file, use the `dsconf import` command.

  ```
  $ dsconf import -h host -p port LDIF_file suffix-DN
  ```

  Using `dsconf import` is slower than using `dsadm import`, but you do not need to stop your server while performing the import operation.

For more detailed information about initializing suffixes, and for examples, see "Initializing a Suffix" on page 52. For detailed command usage, see `dsadm`(1M) and `dsconf`(1M).

5 **(Optional) For each agreement, check that the suffix appears as initialized.**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

▼ To Export a Replicated Suffix to LDIF

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Export the replicated suffix contents in an LDIF file by using one of the following commands:**

  - For an offline export, type:

    ```
    $ dsadm export instance-path suffix-DN LDIF_file
    ```

  - For an online export, type:

    ```
    $ dsconf export -h host -p port suffix-DN LDIF_file
    ```

The following example will export the entire `dc=example,dc=com` replicated suffix and replication information to the file `example_replica_export.ldif`:

```
$ dsconf export -h host2 -p 1389 dc=example,dc=com  \
 /local/dsInst/ldif/example_export_replica.ldif
```

For more information, see "Backing Up to LDIF" on page 225 and the `dsadm(1M)` and `dsconf(1M)` man pages.

Filtering an LDIF File for Fractional Replication

Initializing a replica with fractional replication configured is transparent when using DSCC. Only the selected attributes will be sent to the consumer during the initialization.

If you have configured fractional replication, you should filter out any unused attributes before copying the exported LDIF file to the consumer servers. Directory Server provides the `fildif` tool for this purpose. This tool filters the given LDIF file to keep only the attributes that are allowed by the attribute set defined in your replication agreement.

This tool reads the server's configuration to determine the attribute set definition. To read the configuration file, the `fildif` tool must be run as root or as the user who owns the process and the files (specified by the `nsslapd-localuser` attribute). For example, the following command filters the file exported from the `dc=example,dc=com` suffix in the previous example:

```
$ fildif -i /local/ds1/ldif/example_master.ldif \
 -o /local/ds1/ldif/filtered.ldif -b "cn=host2.example.com:1389, \
 cn=replica,cn=\\"dc=example,dc=com\\",cn=mapping tree,cn=config" -p /local/ds1
```

For the location of the `fildif` command, see "Command Locations" on page 26.

The `-i` and `-o` options are the input and output files, respectively. The `-b` option is the DN of the replication agreement where fractional replication is defined. You can find this DN by using this command:

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement)\
 (nsDS5ReplicaPort=replica-port) (nsDS5ReplicaHost=replica-host))" dn
```

For example:

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement) \
 (nsDS5ReplicaPort=2090)(nsDS5ReplicaHost=host2))" dn
Enter bind password:
version: 1
dn: cn=host2:1389,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
```

For the full command-line syntax for the `fildif` tool, see the `fildif(1)` man page.

You can then use the `filtered.ldif` file produced by `fildif` to initialize the consumer in this replication agreement. Transfer the file to the consumer server and import it as described in "Importing Data From an LDIF File" on page 52.

Initializing a Replicated Suffix by Using Binary Copy

A binary copy enables you to clone an entire server by using the binary backup files from one server to restore the identical directory contents onto another server. You can use a binary copy to initialize or reinitialize any server from the binary copy of a master or hub server, or a consumer from the binary copy of another consumer server.

Note – This advanced procedure interacts with the database files of Directory Server and should only be used by experienced administrators.

Certain restrictions on this feature make it practical and time efficient only for replicas with very large database files, for example, replicas containing millions of entries.

Restrictions for Using Binary Copy With Replication

Because a binary copy moves database files from one machine to another, the mechanism is subject to the following strict limitations:

- Both machines must run the same operating system, including any service packs or patches.
- Both machines must share the same processor architecture. For example, you can perform binary copy between two UltraSPARC® T1 processors but not between an UltraSPARC T1 and an AMD Opteron processor.
- Both machines must be either big endian or little endian.
- Both machines must map memory the same way. For example, you can perform binary copy between server instances on two 64-bit systems, but not between one server instance on a 32-bit system and another on a 64-bit system.
- Both machines must have the same version of Directory Server installed, including binary format (32 bits or 64 bits), service pack, and patch level.
- Both servers must have the same directory tree divided into the same suffixes. The database files for *all* suffixes *must* be copied together. Individual suffixes cannot be copied.
- Each suffix must have the same indexes configured on both servers, including VLV (virtual list view) indexes. The databases for the suffixes must have the same name.
- Each server must have the same suffixes configured as replicas.
- If fractional replication is configured, it must be configured identically on all servers.
- Attribute encryption must not be used on either server.

- The attribute value uniqueness plug-in must have the same configuration on both servers if enabled, and it must be re-configured on the new copy, as explained in the following procedures.

  These procedures describe alternate ways of performing a binary copy: a binary copy that does not require stopping the server and a binary copy that uses the minimum amount of disk space.

Making a Binary Copy for Initializing a Server

This section describes how to make a binary copy for initializing a server, and how to make a binary copy that uses minimum disk space.

▼ To Make a Binary Copy For Initializing a Server

Use this procedure to perform a binary copy for initializing a replicated server because it uses the normal backup functionality to create a copy of the server's database files. Performing a normal backup ensures that all database files are in a coherent state without requiring you to stop the server.

This procedure has certain limitations. The backup and restore operations create copies of the database files on the same machine, thereby doubling the amount of disk space required by those files on each machine. Additionally, the actual copy operation on these files might take a significant amount time if your directory contains gigabytes of data.

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1. **Install Directory Server on the target machine for the new replicated suffix, create a new instance of the server if necessary, and configure the server according to "Restrictions for Using Binary Copy With Replication" on page 271.**

2. **Create all replication agreements in your replication topology that involve this replicated suffix.**

   Include agreements from suppliers to this replica. If this replica is not a dedicated consumer, include agreements from this replica to its consumers. See "Creating and Changing Replication Agreements" on page 263.

3. **Select a fully configured and initialized replica of the same type as you want to initialize, either master, hub, or consumer, and perform a normal backup on it according to "Binary Backup" on page 221.**

4. **Copy or transfer the files from the backup directory to a directory on the target machine by using the `ftp` command, for example.**

5 **If you have initialized a new master in a multimaster replication scenario, follow the procedures in "Restoring a Master in a Multi-Master Scenario" on page 230.**

▼ To Use Binary Copy for Initializing a Server Using Minimum Disk Space

This procedure uses less disk space and takes less time because it does not make backup copies of the database files. However, it requires you to stop the server that is being cloned to order to ensure that the database files are in a coherent state.

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1 **Install Directory Server on the target machine for the new replicated suffix, create a new instance of the server if necessary, and configure the server according to "Restrictions for Using Binary Copy With Replication" on page 271.**

2 **Create all replication agreements in your replication topology that involve this replica.**

Include agreements from suppliers to this replica. If this replica is not a dedicated consumer, include agreements from this replica to its consumers. See "Creating and Changing Replication Agreements" on page 263.

3 **Stop the target server that will be initialized or reinitialized, as described in "Starting, Stopping, and Restarting a Directory Server Instance" on page 45 .**

4 **Select a fully configured and initialized replica of the same type that you want to initialize, either master, hub, or consumer, and stop this server as well.**

If you are cloning a master replica in a multimaster configuration, ensure that it is fully up-to-date with all of the latest changes from the other masters before stopping it.

5 **Restart both the source and the target servers.**

Initializing Replicas in Cascading Replication

In the case of cascading replication, always initialize replicas in the order shown in the following procedure.

▼ To Initialize Replicas in Cascading Replication

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **If you also have multimaster replication, ensure that one master has the complete set of data to replicate, then use this master to initialize the replica on each of the other masters.**

2. **Initialize the replicas on the first-level hub replicas from their master replicas.**
3. **If you have several levels of hubs, initialize each level from the previously initialized level of hubs.**
4. **From the last level of hub replicas, initialize the replicas on the dedicated consumers.**

Indexing Replicated Suffixes

Indexes are not replicated automatically from one server instance to another. To index an attribute for all server instances holding a replicated suffix, perform one of the following actions.

- Manage all server instances holding the replicated suffix as a server group in DSCC. Add the index to one server in the group, then use the Copy Server Configuration action to copy index settings to other servers in the group.

  For more information about DSCC, see "Directory Service Control Center Interface" on page 537.
- Manage the index on each server instance with the `dsconf` command, as described in Chapter 12, "Directory Server Indexing."
- Use binary copy to initialize suffixes, as described in "Initializing a Replicated Suffix by Using Binary Copy" on page 271.

Incrementally Adding Many Entries to Large Replicated Suffixes

If you have a directory with a very large number of entries and you want to add a large quantity of entries, do not use `ldapmodify -a` because it is too time consuming. Instead, add the new entries incrementally by using the `dsconf import` command with an option for adding entries in a replicated topology. When you import the entries, an LDIF file is generated that contains the additions as well as replication metadata. You then import this generated LDIF file to the other replicas. The generated LDIF file ensures that replication synchronization is constant across the replicas to which you add data.

▼ To Add Many Entries to Large Replicated Suffixes

Before You Begin This procedure generates a large LDIF file. Before running the first `dsconf import` command, ensure that you have enough disk space available for the generated LDIF file.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

Caution – This procedure can be used to initialize a server with a large number of entries in several passes. However, if one of the imports fails, the whole database can be lost. Be sure to backup data prior to each import.

1 **On any master replica, import the entries.**

```
$ dsconf import -h host -p port -K generated-LDIF-file suffix-DN
```

The `-K` option ensures that existing data is not removed. It also generates a file *generated-LDIF-file*, that contains the new entries and information required by the replication process.

2 **On all other replicas, import the file generated in the previous step.**

```
$ dsconf import -h host -p port \
-K -f incremental-output=no generated-LDIF-file suffix-DN
```

The option `-f incremental-output=no` specifies that an additional LDIF file will not be generated. Only one generated LDIF file is needed for this procedure.

Replication and Referential Integrity

If you are using the referential integrity plug-in with replication, you must enable it on all master servers. You do not need to enable it on hub or consumer servers.

The following limitations are associated with the use of the referential integrity plug-in in a replication environment:

- The plug-in must be enabled on all servers containing master replicas.
- You must enable the plug-in with the same configuration on every master.
- It is not useful to enable the plug-in on servers containing only hub or consumer replicas.

For information about configuring the referential integrity plug-in, see "To Configure the Referential Integrity Plug-In" on page 252.

Replication Over SSL

You can configure Directory Servers involved in replication so that all replication operations occur over an SSL connection.

▼ To Configure Replication Operations for SSL

This procedure shows example commands for setting up replication on a replication topology with two masters.

Note – This example shows a simple replication configuration, using a self-signed certificate as generated during instance creation. When setting up replication over SSL in a production environment, you will have better security if you use Certificate Authority trusted certificates instead.

Replication over SSL will fail if the supplier server certificate is an SSL server-only certificate that cannot act as a client during an SSL handshake.

While replication is secure by SSL, authentication of the replication manager is still done using a simple bind and password. You can use client-based authentication to fully secure replication, but this requires more complex settings. For more information about replication using client based authentication, see "To Configure Client Authentication Based Replication for SSL" on page 278

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Create new servers and start them.

```
$ dsadm create -p 1389 -P 1636 /local/ds1
$ dsadm create -p 2389 -P 2636 /local/ds2

$ dsadm start /local/ds1
$ dsadm start /local/ds2
```

For more information about configuring SSL, see "Using SSL With Directory Server" on page 102.

2 On all servers, create empty suffixes.

```
$ dsconf create-suffix -e -w password-file -p 1389 dc=example,dc=com
$ dsconf create-suffix -e -w password-file -p 2389 dc=example,dc=com
```

3 On all servers, set the multimaster password file.

```
$ dsconf set-server-prop -e -i -w password-file -h example1.server -p 1389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd1.txt
$ dsconf set-server-prop -e -i -w password-file -h example2.server -p 2389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd2.txt
```

4 On all servers, enable replication.

```
$ dsconf enable-repl -h example1.server -p 1389 -e -i -w password-file \
-d 1 master dc=example,dc=com
$ dsconf enable-repl -h example2.server -p 2389 -e -i -w password-file \
-d 2 master dc=example,dc=com
```

5 **On all servers, export the existing default certificate.**

```
$ dsadm show-cert -F der -o certfile1 /local/ds1 defaultCert
$ dsadm show-cert -F der -o certfile2 /local/ds2 defaultCert
```

6 **On all servers, add the certificate from all other servers.**

```
$ dsadm add-cert --ca /local/ds1 "ds2 Repl Manager Cert" certfile2
$ dsadm add-cert --ca /local/ds2 "ds1 Repl Manager Cert" certfile1
```

7 **Create replication agreement between the servers just configured.**

Note that secure LDAP ports are used for the replication agreements.

```
$ dsconf create-repl-agmt -h example1.server -p 1389 -e -i -w password-file\
 --auth-protocol "ssl-simple" dc=example,dc=com example2.server:2636
$ dsconf create-repl-agmt -h example2.server -p 2389 -e -i -w password-file\
 --auth-protocol "ssl-simple" dc=example,dc=com example1.server:1636
```

8 **For all replication agreements, configure the authentication password file to be the replication manager password file of the consumer (destination) server in the replication agreement.**

```
$ dsconf set-repl-agmt-prop -h example1.server -p 1389 -e -i -w password-file\
 dc=example,dc=com example2.server:2636 auth-pwd-file:/local/ds1/replmanrpwd2.txt
$ dsconf set-repl-agmt-prop -h example2.server -p 2389 -e -i -w password-file\
 dc=example,dc=com example1.server:1636 auth-pwd-file:/local/ds1/replmanrpwd1.txt
```

After you have initialized the suffixes, the supplier will send all replication update messages to the consumer over SSL and will use certificates if you chose that option. Customer initialization will also use a secure connection if performed through DSCC using an agreement configure for SSL.

9 **On all servers, restart the server in order to take configuration changes into account.**

```
$ dsadm restart /local/ds1
$ dsadm restart /local/ds2
```

10 **On one of the master servers, initialize the suffix.**

```
$ dsconf import -h example1.server -p 1389 -e -i \
-w password-file /tmp/Example.ldif dc=example,dc=com
```

11 **On all servers not yet initialized, initialize the servers by using a replication agreement.**

```
$ dsconf init-repl-dest -e -i -w password-file \
-h example1.server -p 1389 dc=example,dc=com example1.server:2636
```

▼ To Configure Client Authentication Based Replication for SSL

In the following procedure, it is assumed that you requested properly signed certificate/key pairs from a trusted Certificate Authority (CA) and the CA certificate of such authority is present in all security databases.

The certificate/key pairs should be issued to the user having replication rights, that is the certificate subject is the DN of a user allowed to transfer replication data between the servers. In the following example, such users are `ou=user1,o=users` and `ou=user1,o=users`; the certificates short names in the security database are `replmgr1` and `replmgr2` respectively.

1 Create new servers.

```
$ dsadm create -p 1389 -P 1636 /local/ds1
$ dsadm create -p 2389 -P 2636 /local/ds2
```

2 Add a user Certificate/Key pair on each server, as received by the CA.

```
$ dsadm import-cert /local/ds1 user1.der
$ dsadm import-cert /local/ds2 user2.der
```

The `user1.der` and `user2.der` are the CA provided files.

3 Export the users' certificates for later use

```
$ dsadm show-cert -F ascii /local/ds1 replmgr1 > user1.ldif
$ dsadm show-cert -F ascii /local/ds2 replmgr2 > user2.ldif
```

The files should contain `base64` encoded binary certificates.

4 Start the servers.

```
$ dsadm start /local/ds1
$ dsadm start /local/ds2
```

5 Create empty suffixes on all the servers, where the users and their certificate will be stored.

```
$ dsconf create-suffix -p 1389 -e o=example.com
$ dsconf create-suffix -p 2389 -e o=example.com
$ dsconf create-suffix -p 1389 -e o=users
$ dsconf create-suffix -p 2389 -e o=users
```

Note – Alternatively, the users and their certificates could be in another suffix. It is not recommended to have the user in the same suffix that is to be replicated.

---

6 On all servers, enable replication.

```
$ dsconf enable-repl -p 1389 -e -d 1 master o=example.com
$ dsconf enable-repl -p 2389 -e -d 1 master o=example.com
```

7 Prepare the users to be set as replication managers. Edit `user1.ldif` and `user2.ldif` to look like the following:

```
dn: cn=user1,o=users
objectclass: top
objectclass: inetorgperson
sn: user1
userCertificate;binary:: MIIBqDCCARGgAwIBAgI <...>
 dXNlcnMwHh <...>
 <...>
```

The files must be a valid LDIF files.

Get rid of the lines, `BEGIN CERTIFICATE` and `END CERTIFICATE`. The value of `userCertificate;binary::` is simply the `base64` encoding. If it spans multiple lines, the first character of the line must be a space.

8 Add the user definitions on the server where the user is going to be allowed to replicate.

```
$ ldapmodify -p 1389 -D binddn -w password -a < user2.ldif
$ ldapmodify -p 2389 -D binddn -w password -a < user1.ldif
```

Note – Alternatively, you can issue the `ldapmodify` commands directly and create the two users interactively. Make sure that you use the correct syntax while setting the `userCertifacte` attribute.

---

9 Set the user allowed to replicate between servers as replication manager.

```
$ dsconf -p 1389 set-suffix-prop repl-manager-bind-dn: cn=user2,o=users
$ dsconf -p 2389 set-suffix-prop repl-manager-bind-dn: cn=user1,o=users
```

10 Set the server certificate to use the user Certificate/key pair as its own.

```
$ dsconf -p 1389 set-server-prop ssl-rsa-cert-name:replmgr1
$ dsconf -p 2389 set-server-prop ssl-rsa-cert-name:replmgr2
```

11 Restart the servers to take into account the new changes.

```
$ dsadm restart /local/ds1
$ dsadm restart /local/ds2
```

12 Create the replication agreements.

```
$ dsconf create-repl-agmt -p 1389 -e -A ssl-client o=example.com hostname:2636
$ dsconf create-repl-agmt -p 2389 -e -A ssl-client o=example.com hostname:1636
```

Replication Over a WAN

Directory Server enables you to perform all forms of replication including multimaster replication between machines connected through a wide area network (WAN). This replication allows supplier servers to initialize and update consumers by making optimal use of the bandwidth over networks with higher latency and lower bandwidth.

Note – When deploying or troubleshooting a replication topology that replicates over a WAN, you must check network speed, latency, and packet loss. Network problems in any of these areas might cause replication delay.

In addition, replication data transfer rates will always be less than what the available physical medium allows in terms of bandwidth. If the update volume between replicas cannot physically be made to fit into the available bandwidth, tuning will not prevent your replicas from diverging under heavy update load. Replication delay and update performance are dependent on many factors, including but not limited to: modification rate, entry size, server hardware, error rates, average latency, and average bandwidth.

If you have questions about replication in your environment, contact your Sun Service Provider.

Internal parameters of the replication mechanism are optimized by default for WANs. However, if you experience slow replication due to the factors mentioned previously, you might want to empirically adjust the window size and group size parameters. You might also be able to schedule your replication to avoid peak network times, thus improving your overall network usage. Finally, Directory Server supports the compression of replication data to optimize bandwidth usage.

Configuring Network Parameters

The window and group network parameters determine how the replication mechanism groups entries to send them more efficiently over the network. These parameters affect how suppliers and consumers exchange replication update messages and acknowledgments. The parameters are configurable in every replication agreement, which allows you to tailor the replication performance according to the specific network conditions of each consumer.

Monitor the effects of any modifications that you make and adjust the parameters accordingly. Refer to "Getting Replication Status" on page 292 for instructions. You do not need to interrupt replication to modify the window size and group size parameters.

Configuring Window Size

The window size (default value `10`) represents the maximum number of update messages that can be sent without immediate acknowledgment from the consumer.

It is more efficient to send many messages in quick succession instead of waiting for an acknowledgment after each message. Using the appropriate window size, you can eliminate the time replicas spend waiting for replication updates or acknowledgments to arrive.

If your consumer replica is lagging behind the supplier, increase the window size to a higher value than the default, such as 100, and check replication performance again before making further adjustments. When the replication update rate is high and the time between updates is therefore small, even replicas connected by a local area network (LAN) can benefit from a higher window size.

▼ To Configure Window Size

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Modify the window size.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port\
 transport-window-size:value
```

For example:

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-window-size:20
```

Configuring Group Size

The group size (default value `1`) represents the maximum number of data modifications that can be bundled into a single update message. If the network connection appears to be impeding replication, increase the group size to a higher value than the default, such as `10`, and recheck replication performance.

When increasing the group size, make sure that the following are true:

- The window size is set significantly higher than the group size.
- The window size divided by the group size is much greater than the value for `nsslapd-maxThreadsPerConn` under `cn=config` on the consumer (typically twice as large).

  When the group size is set higher than 1, the supplier does not wait to fill a group before sending updates to the consumer.

▼ To Configure Group Size

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Modify the group size.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-group-size:value
```

For example:

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-group-size:10
```

Scheduling Replication Activity

If immediate synchronization between your replicas is not critical, you can schedule replication during periods of low network usage. Replication of data should complete significantly faster when the network is more available.

You can schedule replication to start and end at a certain time of day, on a daily or weekly basis. You can do this independently for every consumer through its replication agreement. The new schedule will take effect immediately, causing the next replication of data for the corresponding consumer to be delayed until first allowed by the schedule.

▼ To Schedule Replication Activity

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Modify the replication schedule.**

  ```
  $ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
   host:port repl-schedule:value
  ```

  For example, if you want to set replication to occur between 2:00 and 4:00 every night, type:

  ```
  $ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
   repl-schedule:"0200-0400 0123456"
  ```

  where `0123456` indicate the days of the week, with `0` representing Sunday, `1` representing Monday, and so on.

Configuring Replication Compression

To reduce the bandwidth used by replication, you may configure replication to compress the data that is sent when updating consumers. The replication mechanism uses the Zlib compression library. Both supplier and consumer must be running on a Solaris or Linux platform to enable compression.

You should empirically test and select the compression level that gives you best results for your expected replication usage in your WAN environment. Do not set this parameter in a LAN where there is wide network bandwidth because the compression and decompression computations will slow down replication.

▼ To Configure Replication Compression

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Configure replication compression on the replication agreement entry in the master server.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-compression:level
```

where *level* can be `high`, `medium`, `low`, or `none`.

For example, to use the fastest compression when sending replication updates to the consumer on `host1:1389`, type:

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-compression:high
```

For more information about setting the compression level, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

Modifying the Replication Topology

This section explains these aspects of managing an existing replication topology:

Changing the Replication Manager

You can edit a replication agreement to change the replication manager identity that is used to bind to the consumer server. To avoid any interruption of the replication, you should define the new replication manager entry or certificate entry on the consumer before modifying the replication agreement. However, if replication is interrupted due to a bind failure, the replication mechanism will automatically send all the necessary updates when you correct the error, within the limits of the replication recovery settings. For the procedure, see "Using a Non-Default Replication Manager" on page 261.

Managing Replication Agreements

You can disable, enable, or delete a replication agreement.

Disabling a Replication Agreement

When a replication agreement is disabled, the master stops sending updates to the designated consumer. Replication to that server is stopped, but all settings in the agreement are preserved. You may resume replication by re-enabling the agreement at a later time. See "Enabling a Replication Agreement" on page 284 for information about resuming the replication mechanism after an interruption.

▼ To Disable a Replication Agreement

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Disable a replication agreement.**

```
$ dsconf disable-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

For example:

```
$ dsconf disable-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

Enabling a Replication Agreement

Enabling a replication agreement resumes replication with the designated consumer. However, if replication has been interrupted longer than the replication recovery settings allow and the consumer was not updated by another supplier, you must reinitialize the consumer. The replication recovery settings are the maximum size and age of this supplier's change log and the purge delay of the consumer (see "To Perform Advanced Consumer Configuration" on page 257).

When the interruption is short and replication can recover, the master will update the consumer automatically when the agreement is re-enabled.

▼ To Enable a Replication Agreement

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Enable a replication agreement.**

```
$ dsconf enable-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

For example:

```
$ dsconf enable-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

Deleting a Replication Agreement

Deleting a replication agreement stops the replication to the corresponding consumer and removes all configuration information about the agreement. If you want to resume replication at a later date, disable the agreement instead, as described in "Disabling a Replication Agreement" on page 284.

▼ To Delete a Replication Agreement

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Delete a replication agreement.**

```
$ dsconf delete-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

For example:

```
$ dsconf delete-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

Promoting or Demoting Replicas

Promoting or demoting a replica changes its role in the replication topology. Dedicated consumers can be promoted to hubs, and hubs can be promoted to masters. Masters can be demoted to hubs, and hubs can also be demoted to dedicated consumers. However, masters cannot be demoted directly to consumers, just as consumers cannot be promoted directly to masters.

The allowed promotions and demotions within the multimaster replication mechanism make the topology very flexible. A site that was formerly served by a consumer replica might grow and require a hub with several replicas to handle the load. If the load includes many modifications to the replica contents, the hub can become a master to allow faster local changes that can then be replicated to other masters at other sites.

When promoting or demoting replicas, be aware of the following:

- If you promote a consumer, it becomes a hub. If you promote a hub, it becomes a master. You cannot promote a server directly from consumer to master. You must first promote the consumer to a hub, then promote the hub to a master. Likewise, when demoting a master to a consumer, you must demote the master to a hub before demoting from a hub to a consumer.
- When demoting a master to a hub, the replica will become read-only and be configured for sending referrals to the remaining masters. The new hub will retain all of its consumers, whether hubs or dedicated consumers.

- Demoting a single master to a hub will create a topology without a master replica. Directory Server will allow you to do this under the assumption that you will define a new master. However, it is better to add a new master as a multimaster and allow it to be initialized before demoting the other master.
- Before demoting a hub to a consumer, you must disable or delete all replication agreements to and from the hub. If you do not do this, the demote operation will fail with this error: `LDAP_OPERATIONS_ERROR "Unable to demote a hub to a read-only replica if some agreements are enabled"`.

  If the hub's consumers were not updated by other hubs or masters, they will no longer be updated. You should create new agreements on the remaining hubs or masters to update these consumers.
- When promoting a consumer to a hub, its change log is enabled, and you may define new agreements with consumers.
- When promoting a hub to a master, the replica will accept modification requests, and you may define new agreements with other masters, hubs, or dedicated consumers.

▼ To Promote or Demote a Replica

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Promote or demote a replica by using one of the commands:**

```
dsconf promote-repl [-d REPL_ID] SUFFIX_DN [SUFFIX_DN...]

$ dsconf promote-repl -h host -p port [-d REPL_ID] SUFFIX_DN [SUFFIX_DN...]

$ dsconf demote-repl -h host -p port SUFFIX_DN [SUFFIX_DN...]
```

Disabling a Replicated Suffix

Disabling a replicated suffix removes it from the replication topology. It will no longer be updated or send updates, depending on its role as a master, hub, or consumer. Disabling a suffix on a supplier server deletes all replication agreements, and they will have to be recreated if the replica is enabled again.

▼ To Disable a Replicated Suffix

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Disable a replicated suffix.**

```
$ dsconf disable-repl -h host -p port suffix-DN
```

For example:

```
$ dsconf disable-repl -h host2 -p 1389 dc=example,dc=com
```

Keeping Replicated Suffixes Synchronized

After you stop a Directory Server involved in replication for regular maintenance, when it comes back online, you need to ensure that it gets updated through replication immediately. In the case of a master in a multimaster environment, the directory information needs to be updated by another master in the multimaster set. In other cases, after a hub server or a dedicated consumer server is taken offline for maintenance, when they come back online, they need to be updated by the master server.

This section describes the replication retry algorithm and explains how to force replication updates to occur without waiting for the next retry.

Note – The procedures described in this section can be used only when replication is already set up *and* consumers have been initialized.

Replication Retry Algorithm

When a source replica is unsuccessful in replicating to a destination, it retries periodically in incremental time intervals. The retry intervals depend on the error type.

Note that even if you have configured replication agreements to always keep the source replica and the destination replica synchronized, this is not sufficient to immediately update a replica that has been offline for over five minutes.

▼ To Force Replication Updates

If replication has stopped, you can force replication updates to the destination suffixes.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **On the source server, restart replication updates to the destination server.**

  `$ dsconf update-repl-dest-now -h` *host* `-p` *port* *suffix-DN* *destination-host*`:`*destination-port*

  For example:

  ```
  $ dsconf update-repl-dest-now -h host2 -p 1389 dc=example,dc=com host1:1389
  ```

Moving a Master Replica to a New Machine

In some situations, it might be necessary to move a master replica to a different machine. If you do not need to use the same host name and port number, use `dsconf change-repl-dest` to change the host name and port number of the remote replica. For more information, see "To Change the Destination of a Replication Agreement" on page 264.

If you need to retain the same host name and port number, you must remove the master from the existing topology, and then re-add the master to the topology.

It is much easier to use DSCC to perform these tasks, because DSCC takes care of any impacted replication agreements. If you use DSCC, however, you cannot specify the same replica ID that the master originally had in the topology. To use the same replica ID, you must use the command line to perform these tasks, as follows.

▼ To Remove a Master From an Existing Replication Topology

Before You Begin Make sure that all changes from the master have already been replicated.

1. **If you can, back up the master using binary copy so that you do not lose any changes.**
2. **Demote the master replica to a hub replica.**

   See "Promoting or Demoting Replicas" on page 285.
3. **Wait for the hub to start replicating to other servers.**

   When the hub opens a replication session to the other servers in the topology, it remains in the RUV but is no longer used in referrals.
4. **Stop the hub.**

   See "Starting, Stopping, and Restarting a Directory Server Instance" on page 45.
5. **Remove the hub from the topology.**

   See "Disabling a Replicated Suffix" on page 286.

▼ To Add a Master to an Existing Replication Topology

1. **Add the master replica, using the same replica ID.**

   See "Enabling Replication on a Master Replica" on page 260.
2. **Recreate the replication agreements from that master to the other replicas in the topology.**
3. **Initialize the new master.**

   a. **If you were able to back up the master, initialize the master from this backup.**

b. If you were not able to back up the master (in the event of a machine crash), initialize the master from another master in the topology.

Replication With Releases Prior to Directory Server 7.0

This section provides information about how to configure replication with releases of Directory Server prior to 7.0.

Replicating Between Directory Server 7.0 and Directory Server 6 or 5.2

Directory Server 7.0, 6, and 5.2 are compatible with regard to replication configuration, with the following exceptions:

- Replication priority is not supported in version 5.2. If you configure replication priority on a 7.0 master replica, the replication priority will be transferred to consumers running Directory Server 7.0, but not to any consumers running a previous version of Directory Server.
- An unlimited number of masters is not supported on replication topologies that contain Directory Server 5.2 masters. Although Directory Server 7.0 supports an unlimited number of masters in a replication topology, this number is limited to 4 if your replication topology includes any Directory Server 5.2 master servers.

Using the Retro Change Log

The retro change log is used by LDAP clients to ascertain the history of changes made to the Directory Server data. The retro change log is stored in a separate database to the Directory Server change log, under the suffix `cn=changelog`.

A retro change log can be enabled on a standalone server or on each server in a replication topology. When the retro change log is enabled on a server, by default updates to all suffixes on that server are logged. The retro change log can be configured to log updates to specified suffixes only.

For information about using the retro change log in a replicated topology and about restrictions on using the retro change log, see "Replication and the Retro Change Log Plug-In" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

For information about the attributes of an entry in the retro change log, see the `changeLogEntry(5dsoc)` man page.

For more information about modifying the retro change log, see the `dsconf(1M)` man page.

This section explains various ways that you can use the retro change log.

To Enable the Retro Change Log

To use the retro change log, you must enable it.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Modify the retro change log configuration entry:**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-enabled:on
   ```

2. **Restart the server.**

   For information, see "Starting, Stopping, and Restarting a Directory Server Instance" on page 45.

To Configure the Retro Change Log to Record Updates for Specified Suffixes

When the retro change log is enabled on a server, by default it records updates to all suffixes on the server. This procedure describes how to configure the retro change log to record updates to specified suffixes only.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Modify the retro change log configuration entry:**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-suffix-dn:suffix-DN
   ```

   For example, to log changes only on the `cn=Contractors,dc=example,dc=com` suffix and the `ou=People,dc=example,dc=com` suffix, use this command:

   ```
   $ dsconf set-server-prop -h host2 -p 1389 \
    retro-cl-suffix-dn:"cn=Contractors,dc=example,dc=com" \
    retro-cl-suffix-dn:"ou=People,dc=example,dc=com"
   ```

   To add a suffix to an existing list of specified suffixes, use this command:

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-suffix-dn+:suffix-DN
   ```

2. **Restart the server.**

   For information, see "Starting, Stopping, and Restarting a Directory Server Instance" on page 45.

To Configure the Retro Change Log to Record Attributes of a Deleted Entry

This procedure describes how to configure the retro change log to record specified attributes of an entry when that entry is deleted.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Specify the attributes that must be recorded:**

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr: \
 attribute1 attribute2
```

For example, to set the retro change log to record the UID attributes of deleted entries, use this command:

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr:uid
```

To add an attribute to an existing list of specified attributes, use this command:

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr+:attribute
```

2 **Restart the server.**

For information, see "Starting, Stopping, and Restarting a Directory Server Instance" on page 45.

To Trim the Retro Change Log

The entries in the retro change log can be removed automatically after a specified period of time. To configure the period of time after which entries are deleted automatically, make sure that the retro change log is enabled, then set the `nsslapd-changelogmaxage` configuration attribute in the `cn=Retro Changelog Plugin, cn=plugins, cn=config` entry.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Check that the retro change log is enabled.**

```
$ dsconf get-server-prop -h host -p port retro-cl-enabled
```

2 **If the retro change log is not enabled, enable it.**

```
$ dsconf set-server-prop -h host -p port retro-cl-enabled:on
```

3 **Set the maximum age for changes logged.**

```
$ dsconf set-server-prop -h host -p port retro-cl-max-age:duration
```

where *duration* can be either `undefined` (no age limit) or one of the following:

- `s` for seconds
- `m` for minutes
- `h` for hours
- `d` for days
- `w` for weeks

For example, to set the retro change log maximum age to two days, type:

```
$ dsconf set-server-prop -h host 2 -p 1389 retro-cl-max-age:2d
```

Entries that exceed this age are trimmed from the change log every 5 minutes.

Access Control and the Retro Change Log

The retro change log supports search operations. It is optimized for searches that include filters of this form:

```
(&(changeNumber>=X)(changeNumber<=Y))
```

As a general rule, do not perform add or modify operations on the retro change log entries. You can delete entries to trim the size of the log. The only time that you need to perform a modify operation on the retro change log is to modify the default access control policy.

When the retro change log is created, by default, the following access control rules apply:

- Read, search, and compare rights are granted to Directory Manager.
- Write and delete access are not granted, except implicitly to the Directory Manager.

  Do not grant read access to anonymous users because the retro change log entries can contain modifications to sensitive information such as passwords. You may want to further restrict access to the retro change log contents if authenticated users should not be allowed to view its contents.

To modify the default access control policy that applies to the retro change log, modify the `aci` attribute of the `cn=changelog` entry. Refer to Chapter 6, “Directory Server Access Control.”

Getting Replication Status

You can get replication status by using DSCC or by using command-line tools.

Getting Replication Status in DSCC

You can graphically view replication, including replication agreements and replication delay, by using the Suffix tab. For more information, see the DSCC online help.

In addition, you can use DSCC to view your replication topology, as shown in following figure.

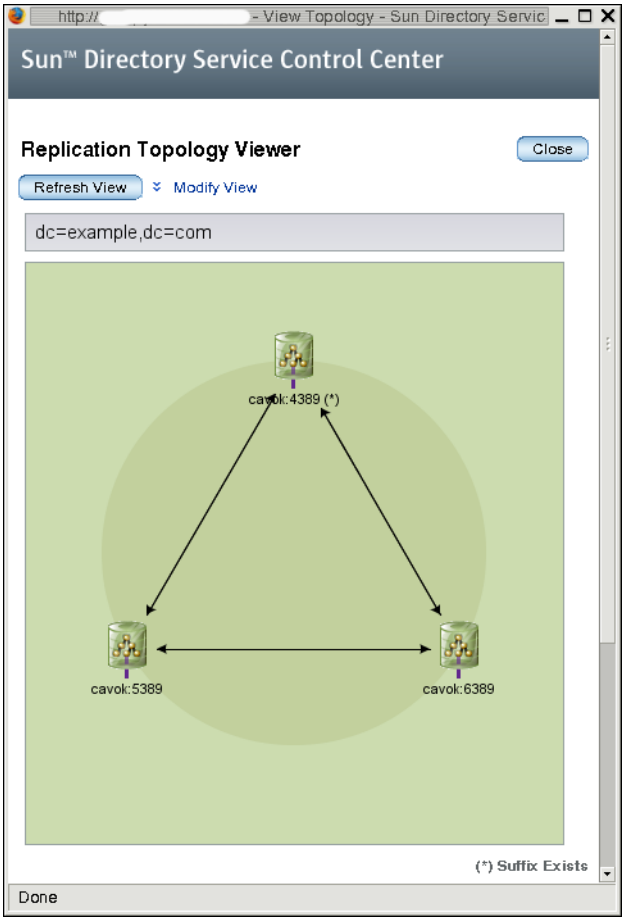


FIGURE 10–1 Sample Replication Topology

Getting Replication Status by Using the Command Line

If you are unable to use DSCC, use command-line tools to obtain information about your replication deployment.

The man pages provide full command-line syntax and usage examples for these tools.

- `repldisc` - "Discovers" and constructs a table of all known servers in a replication deployment. See the `repldisc(1)` man page.
- `insync` - Indicates the synchronization state between a supplier and one or more consumer replicas. See the `insync(1)` man page.
- `entrycmp` - Compares the same entry in two or more replicas. See the `entrycmp(1)` man page.

To find the directories where these commands are located, see "Command Locations" on page 26.

Solving Common Replication Conflicts

Multimaster replication uses a loose consistency replication model. This means that the same entries may be modified simultaneously on different servers. When updates are sent between the two servers, any conflicting changes must be resolved. Most resolution occurs automatically. For example, the timestamp associated with the change on each server is resolved by the most recent change taking precedence. However, some change conflicts require manual intervention to reach a resolution.

This section covers the following topics:

Solving Replication Conflicts by Using DSCC

The easiest way to resolve a replication conflict is by using DSCC. See the DSCC online help for information.

Solving Replication Conflicts by Using the Command Line

You can solve replication conflicts by using the command line. Entries that have a change conflict that cannot be resolved automatically by the replication process contain the operational attribute `nsds5ReplConflict` as a conflict marker.

To find entries with conflicts, periodically search for entries that contain this attribute. For example, you could use the following `ldapsearch` command to find entries with conflicts:

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config \
 -w - -b "dc=example,dc=com" "(nsds5ReplConflict=*)"
```

Note that the `nsds5ReplConflict` attribute is indexed by default.

Solving Naming Conflicts

Entries with identical DNs may be created on separate masters if they are created before the servers replicate the changes to each other. Upon replication, the conflict resolution mechanism will automatically rename the second entry created.

An entry with a DN naming conflict is renamed by including its unique identifier, provided by the operational attribute `nsuniqueid`, in its DN.

For example, if the entry `uid=bjensen,ou=People,dc=example,dc=com` is created simultaneously on two masters, both will have the following two entries after replication:

- `uid=bjensen,ou=People,dc=example,dc=com`
- `nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com`

The second entry must be given a useful DN. You can delete the conflicting entry and add it again with a non-conflicting name. However, renaming the entry ensures that its contents have not changed. The renaming procedure depends on whether the naming attribute is single-valued or multivalued. See the following procedures.

▼ To Rename a Conflicting Entry That has a Multivalued Naming Attribute

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Rename the entry while keeping the old RDN value, for example:

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com
changetype: modrdn
```

```
newrdn: uid=bj66446001
deleteoldrdn: 0
^D
```

You cannot delete the old RDN value in this step because it also contains the `nsuniqueid` operational attribute, which cannot be deleted.

2 Remove the old RDN value of the naming attribute and the conflict marker attribute, for example:

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bj66446001,dc=example,dc=com
changetype: modify
delete: uid
uid: bjensen
-
delete: nsds5ReplConflict
^D
```

▼ To Rename a Conflicting Entry With a Single-Valued Naming Attribute

When the naming attribute in a duplicate entry is single-valued, for example `dc` (domain component), you cannot simply rename the entry to another value of the same attribute. Instead, you must give the entry a temporary name.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Rename the entry by using a different naming attribute, and keep the old RDN, for example:

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+dc=HR,dc=example,dc=com
changetype: modrdn
newrdn: o=TempHREntry
deleteoldrdn: 0
^D
```

You cannot delete the old RDN value in this step because it also contains the `nsuniqueid` operational attribute, which cannot be deleted.

2 Change the desired naming attribute to a unique value and remove the conflict marker attribute, for example:

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: o=TempHREntry,dc=example,dc=com
changetype: modify
replace: dc
```

```
dc: NewHR
delete: nsds5ReplConflict
^D
```

3 Rename the entry back to the intended naming attribute, for example:

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=NewHR,dc=example,dc=com
changetype: modrdn
newrdn: dc=HR
deleteoldrdn: 1
^D
```

By setting the value of the `deleteoldrdn` attribute to `1`, you delete the temporary attribute-value pair `o=TempHREntry`. If you want to keep this attribute, set the value of the `deleteoldrdn` attribute to `0`.

Solving Orphan Entry Conflicts

When a delete operation is replicated, and the consumer server finds that the entry to be deleted has child entries, the conflict resolution procedure creates a *glue* entry to avoid having orphaned entries in the directory.

In the same way, when an add operation is replicated, and the consumer server cannot find the parent entry, the conflict resolution procedure creates a glue entry representing the parent so that the new entry is not an orphan entry.

Glue entries are temporary entries that include the object classes `glue` and `extensibleObject`. Glue entries can be created in various ways:

- If the conflict resolution procedure finds a deleted entry with a matching unique identifier, the glue entry is a resurrection of that entry. It also includes the `glue` object class and the `nsds5ReplConflict` attribute.

  In such cases, you can either modify the glue entry to remove the `glue` object class and the `nsds5ReplConflict` attribute to keep the entry as a normal entry, or delete the glue entry and its child entries.
- The server creates a minimal entry with the `glue` and `extensibleObject` object classes.

  In such cases, you must either modify the entry to turn it into a meaningful entry or delete the entry and all of its child entries.

Solving Potential Interoperability Problems

For interoperability with applications that rely on attribute uniqueness, such as a mail server, you might need to restrict access to the entries that contain the `nsds5ReplConflict` attribute. If you do not restrict access to these entries, the applications that require only one attribute will pick up both the original entry and the conflict resolution entry that contains the `nsds5ReplConflict` and operations will fail.

To restrict access, you need to modify the default ACI that grants anonymous read access using the following command:

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=example,dc=com
changetype: modify
delete: aci
aci: (target ="ldap:///dc=example,dc=com")
 (targetattr !="userPassword"
 (version 3.0;acl "Anonymous read-search  access";
 allow (read, search, compare)(userdn = "ldap:///anyone");)
-
add: aci
aci: (target="ldap:///dc=example,dc=com")
 (targetattr!="userPassword")
 (targetfilter="(!(nsds5ReplConflict=*))")(version 3.0;acl
 "Anonymous read-search access";allow (read, search, compare)
 (userdn="ldap:///anyone");)
^D
```

The new ACI will keep entries that contain the `nsds5ReplConflict` attribute from being returned in search results.

◆ ◆ ◆ CHAPTER 11

Directory Server Schema

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 might need to extend the schema by creating new object classes and attributes. For an overview of the standard schema and for instructions on designing schema to suit your deployment, see the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

This chapter describes how to manage your schema and covers the following topics:

Managing Schema Checking

When schema checking is on, Directory Server ensures that all import, add, and modify operations conform to the currently defined directory schema.

- The object classes and attributes of each entry conform to the schema.
- The entry contains all required attributes for all of its defined object classes.
- The entry contains only attributes that are allowed by its object classes.

Note – When modifying an entry, Directory Server performs schema checking on the entire entry, not just on the attributes that are being modified. Therefore, the operation might fail if any object class or attribute in the entry does not conform to the schema.

However, schema checking does not verify the validity of attribute values with regard to their syntax.

Schema checking is turned on by default. In general, run Directory Server with schema checking turned on. Many client applications assume that having schema checking turned on is an indication that all entries conform to the schema. However, turning on schema checking does not cause Directory Server to verify the existing contents in the directory. The only way to guarantee that all directory contents conform to the schema is to turn on schema checking before adding any entries or reinitializing all entries.

One case where you might want to turn schema checking off is to accelerate import operations of LDIF files known to conform to the schema. However, there is a risk of importing entries that do not conform to the schema. If schema checking is off, imported entries that do not confirm to the schema are not detected.

See "Replicating Directory Schema" on page 316 for details on using schema checking in replicated environments.

▼ To Fix Schema Compliance Problems

When an entry does not conform to the schema, it might not be possible to search for this entry, and modification operations on the entry might fail. Follow the steps in this procedure to correct the problem.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin To avoid having to fix schema compliance issues, plan your schema ahead of your deployment to minimize schema changes. For more information, see the *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

1. **To determine why the entry does not comply, retrieve the entry and manually compare it with the currently defined schema.**

   See "To View Attribute Types" on page 311 and "To View an Object Class" on page 315 for details.

2. **Modify the entry so that it complies with the schema, or modify the schema so that it complies with the entry.**

Extending Directory Server Schema

When you add new attributes to your schema, you must create a new object class to contain the new attributes. Although it might seem convenient to just add the attributes to an existing object class that already contains most of the attributes you require, doing so compromises interoperability with LDAP clients.

Interoperability of Directory Server with existing LDAP clients relies on the standard LDAP schema. If you change the standard schema, you will also have difficulties when upgrading your server. For the same reasons, you cannot delete standard schema elements. For more information about general guidelines to customize schema, refer to "About Custom Schema" on page 306.

Directory Server schema are stored in attributes of the `cn=schema` entry. Like the configuration entry, this is an LDAP view of the schema that is read from files during server startup.

The method that you use to extend Directory Server schema depends on whether you want control over the file name where schema extensions are stored. It also depends whether you want to push changes to consumers through replication. See the following table to determine which procedure to follow in your specific case.

TABLE 11–1 Ways to Extend Schema

Task	Instructions
You intend to extend the schema through LDAP.	"To Extend Schema Through LDAP" on page 302
You do not use replication. You intend to extend the schema by adding a custom schema file.	"To Extend Schema With a Custom Schema File" on page 303
You use replication. You intend to preserve the file name of your custom schema file on all servers.	"To Extend Schema With a Custom Schema File" on page 303
You use replication. You intend to extend the schema by adding a custom schema file on a master replica. You then let the replication mechanism copy the schema extensions to consumer servers.	"To Extend Schema Using a Schema File and Replication" on page 305

For more information about object classes, attributes, and the directory schema as well as guidelines for extending your schema, see "Designing a Directory Schema" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*. For information about standard attributes and object classes, see *Sun Directory Server Enterprise Edition 7.0 Man Page Reference*.

This section provides information about the various methods to extend the directory schema.

Extending Schema Through LDAP

Because the schema is defined by the LDAP view in `cn=schema`, you can view and modify the schema online using the `ldapsearch` and `ldapmodify` utilities. However, you can modify only schema elements that have the value '`user defined`' for the `X-ORIGIN` field. The server refuses any modification to the other definitions.

New element definitions, and changes that you make to user-defined elements, are saved in the file `99user.ldif`.

▼ To Extend Schema Through LDAP

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Modifying schema definitions from the command line is prone to error because of the long values that you must type exactly. However, you can use this functionality in scripts that need to update your directory schema.

1. **Use the `ldapmodify(1)` command to add or delete individual `attributeTypes` attribute values.**

   See "To Create an Attribute Type" on page 310 or "To Delete Attribute Types" on page 312 for details.

2. **Use the `ldapmodify(1)` command to add or delete individual `objectClasses` attribute values.**

   See "To Create an Object Class" on page 314 or "To Delete an Object Class" on page 316 for details.

See Also To modify one of the values, you must delete the specific value and then add the value as a new value. This process is required because the attributes are multivalued. For details, see "Modifying One Value of a Multi Valued Attribute" on page 85.

Extending Schema With a Custom Schema File

Caution – To extend schema, modifying schema files is not recommended as this method of extending schema is error-prone. To make any changes to the Directory Serverschema, use the `ldapmodify` command, which is more reliable way to extend schema.

Schema files are LDIF files that are located in *instance-path*`/config/schema/`. The *instance-path* corresponds to the file system directory where the Directory Server instance resides. For example, the instance might be located in `/local/dsInst/`. The files define

standard schema that are used by Directory Server and all servers that rely on Directory Server. The files and the standard schema are described in *Sun Directory Server Enterprise Edition 7.0 Reference* and *Sun Directory Server Enterprise Edition 7.0 Man Page Reference*.

Schema files are read once only at startup by the server. The LDIF contents of the files are added to the in-memory LDAP view of the schema in `cn=schema` . Because the order of schema definitions is important, schema file names are prepended with a number and loaded in alphanumerical order. Schema files in this directory are writable only by the system user who is defined during installation.

When defining the schema directly in an LDIF file, do not use the value '`user defined`' for the `X-ORIGIN` field. This value is reserved for schema elements that are defined through the LDAP view of `cn=schema` and that appear in the file `99user.ldif`.

The `99user.ldif` file contains additional ACIs for the `cn=schema` entry and all schema definitions that have been added from the command-line or using DSCC. The `99user.ldif` file is overwritten when new schema definitions are added. If you want to modify this file, you must restart the server immediately to ensure that your changes are current.

Do not modify the standard schema that is defined in the other schema files. You can, however, add new files to define new attributes and object classes. For example, to define new schema elements in many servers, you could define the elements in a file named `98mySchema.ldif` and copy this file to the schema directory on all servers. You would then restart all servers to load your new schema file.

▼ To Extend Schema With a Custom Schema File

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Create your own schema definition file, such as `98mySchema.ldif`.**

The syntax of definitions in the schema files is described in RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`).

Read "When Creating Custom Schema Files" on page 304 before creating custom schema files.

2 **(Optional) If this server is a master replica that sends updates to other servers, copy your schema definition file to each server instance in the replication topology.**

The replication mechanism cannot detect any changes that you make directly to the LDIF files that contain the schema. Therefore, your changes are not replicated to consumers even after restarting the master.

3 **Restart each Directory Server instance to which you copied your schema definition file.**

Your changes take effect when the servers restart and thus reload schema definitions.

When Creating Custom Schema Files

Keep the following in mind when creating custom schema files, especially when you are using replication:

- When adding new schema elements, all attributes must be defined before they can be used in an object class. You can define attributes and object classes in the same schema file.
- Each custom attribute or object class that you create should be defined in only one schema file. This practice prevents the server from overriding any previous definitions when the server loads the most recently created schema. Directory Server loads the schema files in numerical order first, then in alphabetical order.
- When defining new schema definitions manually, best practice is generally to add these definitions to the `99user.ldif` file.

  When you update schema elements using LDAP, the new elements are written automatically to the `99user.ldif` file. As a result, any other schema definition changes that you made in custom schema files might be overwritten. Using only the `99user.ldif` file prevents possible duplications of schema elements and the danger of schema changes being overwritten.
- As Directory Server loads schema files in alphanumeric order with numbers loaded first, you should name custom schema files as follows:

  `[00-99]` *filename*`.ldif`

  The number is higher than any directory standard schema already defined.

  If you name your schema file with a number that is lower than the standard schema files, the server might encounter errors when loading the schema. In addition, all standard attributes and object classes are loaded only after your custom schema elements have been loaded.
- Make sure that custom schema file names are not numerically or alphabetically higher than `99user.ldif` because Directory Server uses the highest sequenced file for its internal schema management.

  For example, if you created a schema file and named it `99zzz.ldif`, the next time you update the schema, all of the attributes with an `X-ORIGIN` value of `'user defined'` would be written to `99zzz.ldif`. The result would be two LDIF files that contain duplicate information, and some information in the `99zzz.ldif` file might be erased.

- As a general rule, identify the custom schema elements that you are adding with the following two items:
  - `'user defined'` in the `X-ORIGIN` field of custom schema files,
  - A more descriptive label such as `'Example.com Corporation defined'` in the `X-ORIGIN` field, so that the custom schema element is easy for other administrators to understand. For example `X-ORIGIN ('user defined' 'Example.com Corporation defined')`.

    If you are adding schema elements manually and do not use `'user defined'` in the `X-ORIGIN` field, the schema elements appear read-only in DSCC.

    The `'user defined'` value is added automatically by the server if you add custom schema definitions by using LDAP or DSCC. However, if you do not add a more descriptive value in the `X-ORIGIN` field, you might later have difficulty understanding what the schema relates to.

  These changes are not replicated automatically. To replicate, you must propagate custom schema files manually to all your servers.

  When you change the directory schema, the server keeps a timestamp of when the schema was changed. At the beginning of each replication session the server compares its timestamp with its consumer's timestamp and, if necessary, pushes any schema changes. For custom schema files, the server maintains only one timestamp, which is associated with the `99user.ldif` file. This means that any custom schema file changes or additions that you make to files other than `99user.ldif` will not be replicated. Therefore, you must propagate custom schema files to all other servers to ensure that all schema information is present throughout the topology.

Extending Schema Using a Schema File and Replication

For information about custom schema files, see "Extending Schema With a Custom Schema File" on page 302. The following procedure explains how to use the replication mechanism to propagate schema extensions to all the servers in a topology.

▼ To Extend Schema Using a Schema File and Replication

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1 Prepare your schema extensions in one of the following ways:

- **Create your own schema definition file, such as `98mySchema.ldif`.**

- **Add your schema extensions to `99user.ldif`.**

The syntax of definitions in the schema files is described in RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`).

2. **Run the `dsadm start` or `dsadm restart` command with `--schema-push` on the master server where you put the schema definition file.**

   This script does not actually push the schema to replicas. Instead the script writes a special attribute into the schema files so that the schema files are replicated as soon as they are loaded. For more information, see the `dsadm`(1M) man page.

3. **Restart the master server where you put the schema definition file.**

   The replication mechanism cannot detect any changes that you make directly to the LDIF files that contain the schema. When you restart the server, the server loads all schema files and then the replication mechanism replicates the new schema to consumers.

About Custom Schema

You can extend the standard schema if it is too limited for your directory needs. Follow these guidelines when customizing schema:

- Reuse existing schema elements whenever possible.
- Minimize the number of mandatory attributes that you define for each object class.
- Do not define more than one object class or attribute for the same purpose.
- Keep the schema as simple as possible.

When customizing the schema, do not modify, delete, or replace any existing definitions of attributes or object classes in the standard schema. Doing so can lead to compatibility problems with other directories and with LDAP client applications.

Do not modify any Directory Server internal operational attributes. You can, however, create your own operational variables for external applications.

Always define object classes instead of using `objectClass: extensibleObject`. Directory Server does not perform schema checking for entries that have the object class `extensibleObject`, so it does not constrain or check what attributes are present on the entry. Typos in applications, for example, `giveName` for the `givenName` attribute type, go unnoticed by Directory Server. Also, Directory Server must assume that all otherwise undefined attributes of `extensibleObject` entries are multivalued and have case-insensitive string syntax. Furthermore, some applications rely on entries having a particular object class. In general, if you have an application that requires an extension to an object class, do not relinquish schema management. Instead, create an auxiliary object class that contains the attributes that are required for the application.

This section contains information about the default directory schema, and about creating customized attributes and object classes.

Default Directory Server Schema

The schema provided with Directory Server is described in a set of files that are stored in the *instance-path*`/config/schema/`directory.

This directory contains all of the common schema for Directory Server and related products. The LDAP v3 standard user and organization schema is located in the `00core.ldif` file. The configuration schema used by earlier versions of the directory is located in the `50ns-directory.ldif` file. The user created elements such as objectclasses and attributes are stored in `99user.ldif`.

Note – Do not modify files in this directory. To manage Directory Server schema, use the `ldapmodify(1)` command.

Object Identifiers

Each LDAP object class or attribute must be assigned a unique name and object identifier (OID). When you define a schema, you need an OID that is unique to your organization. One OID is enough to meet all of your schema needs. You then add new branches on that OID for your attributes and object classes.

Obtaining and assigning OIDs in your schema involves doing the following:

- Obtaining an OID for your organization from the Internet Assigned Numbers Authority (IANA) or a national organization.

  In some countries, corporations already have OIDs assigned to them. If your organization does not already have an OID, you can obtain an OID from IANA.

- Creating an OID registry so you can track OID assignments.

  An OID registry is a list that you maintain, which gives the OIDs and OID descriptions that are used in your directory schema. A OID registry ensures that no OID is ever used for more than one purpose.

- Creating branches in the OID tree to accommodate schema elements.

  Create at least two branches under the OID branch or your directory schema, using *OID*`.1` for attributes and *OID*`.2` for object classes. If you want to define your own matching rules or controls, you can add new branches as needed, such as *OID*`.3`.

Naming Attributes and Object Classes

When creating names for new attributes and object classes, make the name meaningful so your schema is easier to use.

Avoid naming collisions between custom schema elements and existing schema elements by including a unique prefix on custom elements. For example, Example.com Corporation might add the prefix `Example` before each of its custom schema elements. It might also add a special object class called `ExamplePerson` to identify Example.com employees in its directory.

Note that in LDAP, attribute type names and object class names are *case insensitive*. Applications should treat them as case insensitive strings.

When Defining New Object Classes

Add new object classes when the existing object classes do not support all of the information you need to store in a directory entry.

There are two approaches to creating new object classes:

- Create many new object classes, one for each object class structure to which you want to add an attribute.
- Create a single object class that supports all of the attributes that you create for your directory. You create this kind of an object class by defining it to be an `AUXILIARY` object class.

  Suppose your site wants to create the attributes `ExampleDepartmentNumber` and `ExampleEmergencyPhoneNumber`. You can create several object classes that allow some subset of these attributes. You can create an object class called `ExamplePerson` and have it allow the `ExampleDepartmentNumber` and `ExampleEmergencyPhoneNumber` attributes. The parent of `ExamplePerson` would be `inetOrgPerson`. You can then create an object class called `ExampleOrganization` and have it also allow the `ExampleDepartmentNumber` and `ExampleEmergencyPhoneNumber` attributes. The parent of `ExampleOrganization` would be the `organization` object class.

  Your new object classes would appear in LDAP v3 schema format as follows:

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.3 NAME 'ExamplePerson'
 DESC 'Example Person Object Class' SUP inetorgPerson STRUCTURAL MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.4 NAME
 'ExampleOrganization' DESC 'Example Organization Object Class' SUP
 organization STRUCTURAL MAY (ExampleDepartmentNumber
 $ ExampleEmergencyPhoneNumber) )
```

  Alternatively, you can create a single object class that allows all of these attributes. Then you can use the object class with any entry on which you want to use the attributes. The single object class would appear as follows:

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.5 NAME 'ExampleEntry'
 DESC 'Example Auxiliary Object Class' SUP top AUXILIARY MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
```

The new `ExampleEntry` object class is marked `AUXILIARY`, meaning that it can be used with any entry regardless of its structural object class.

Consider the following when deciding how to implement new object classes.

- Multiple `STRUCTURAL` object classes result in more schema elements to create and maintain.

  Generally, the number of elements remains small and needs little maintenance. However, if you plan to add more than two or three object classes to your schema, you might find it easier to use a single object class.
- Multiple `STRUCTURAL` object classes require more careful and more rigid data design.

  Rigid data design forces you to consider the object class structure on which every piece of data is placed. You might find this restriction to be either helpful or cumbersome.
- Single `AUXILIARY` object classes simplify data design when you have data that you want to put on more than one type of object class structure.

  For example, suppose that you want `preferredOS` on both a person and a group entry. You might want to create only a single object class to allow this attribute.
- Design object classes that relate to real objects and group elements that constitute sensible groupings.
- Avoid required attributes for new object classes.

  Requiring attributes can make your schema inflexible. When you create a new object class, allow rather than require attributes.

  After defining a new object class, you need to decide which attributes the object class allows and requires, and from which object class or classes it inherits.

When Defining New Attributes

Add new attributes when the existing attributes do not support all of the information you need to store in a directory entry. Try to use standard attributes whenever possible. Search the attributes that already exist in the default directory schema and use those attributes in association with a new object class.

For example, you might find that you want to store more information on a person entry than the `person`, `organizationalPerson`, or `inetOrgPerson` object classes support. If you want to store birth dates in your directory, no attribute exists within the standard Directory Server schema. You can create a new attribute called `dateOfBirth`. Allow this attribute to be used on entries that represent people by defining a new auxiliary class that allows this attribute.

Managing Attribute Types Over LDAP

This section explains how to create, view, and delete attribute types over LDAP.

Creating Attribute Types

The `cn=schema` entry has a multivalued attribute, `attributeTypes`, that contains definitions of each attribute type in the directory schema. You can add to those definitions by using the `ldapmodify(1)` command.

New attribute type definitions, and changes that you make to user-defined attribute types, are saved in the file `99user.ldif`.

For each attribute type definition, you must provide at least an OID to define your new attribute type. Consider using at least the following elements for new attribute types:

- **Attribute OID.** Corresponds to the object identifier for your attribute. An OID is a string, usually of dotted decimal numbers, that uniquely identifies the schema object.

  For strict LDAP v3 compliance, you must provide a valid numeric OID. To learn more about OIDs or to request a prefix for your enterprise, send email to the IANA (Internet Assigned Number Authority) at `iana@iana.org`, or see the IANA web site (`http://www.iana.org`).
- **Attribute name.** Corresponds to a unique name for the attribute. Also called its attribute type. Attribute names must begin with a letter and contain only ASCII letters, digits, and hyphens.

  An attribute name can contain uppercase letters, but no LDAP client should rely on case to differentiate attributes. Attribute names must be handled in a case-insensitive manner according to section 2.5 of RFC 4512 (`http://www.ietf.org/rfc/rfc4512.txt`).

  You can optionally include alternate attribute names, also referred to as aliases, for your attribute type.
- **Attribute description.** Is short descriptive text that explains the attribute's purpose.
- **Syntax.** Is referenced by the OID and describes the data to be held by the attribute.

  Attribute syntaxes with their OIDs are listed in RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`).
- **Number of values allowed.** By default, attributes are multivalued, but you might want to restrict an attribute to a single value.

▼ To Create an Attribute Type

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Prepare your attribute type definition according to the syntax specified in RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`).**

2. **Use the `ldapmodify(1)` command to add your attribute type definition.**

   Notice that Directory Server adds `X-ORIGIN 'user defined'` to the definition that you provide.

Example 11–1 Creating an Attribute Type

The following example adds a new attribute type with Directory String syntax using the `ldapmodify` command:

```
$ cat blogURL.ldif
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7
 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE )

$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogURL.ldif
Enter bind password:
modifying entry cn=schema

$
```

In a production environment, you would provide a valid, unique OID, not `1.2.3.4.5.6.7`.

Viewing Attribute Types

The `cn=schema` entry has a multivalued attribute, `attributeTypes`, that contains definitions of each attribute type in the directory schema. You can read those definitions by using the `ldapsearch(1)` command.

▼ To View Attribute Types

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Use the `ldapsearch` command to view all the attribute type definitions that currently exist in your directory schema.**

Example 11–2 Viewing Attribute Types

The following command displays definitions for all attribute types:

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes
```

The `-T` option prevents the `ldapsearch` command from folding LDIF lines, so you can more easily work with the output using commands such as `grep` or `sed`. If you then pipe the output of this command through the `grep` command, you can view only the user-defined extensions to directory schema. For example:

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes | grep "user defined"
 attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
```

Deleting Attribute Types

The `cn=schema` entry has a multivalued attribute, `attributeTypes`, that contains definitions of each attribute type in the directory schema. You can delete definitions with `X-ORIGIN 'user defined'` by using the `ldapmodify(1)` command.

Because the schema is defined by the LDAP view in `cn=schema`, you can view and modify the schema online using the `ldapsearch` and `ldapmodify` utilities. However, you can delete only schema elements that have the value '`user defined`' for the `X-ORIGIN` field. The server will not delete other definitions.

Changes that you make to user-defined attributes are saved in the file `99user.ldif`.

▼ To Delete Attribute Types

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **View the definition of the attribute type to delete.**

   See "To View Attribute Types" on page 311 for details.

2. **Use the `ldapmodify(1)` command to delete the attribute type definition as it appears in the schema.**

Example 11–3 Deleting an Attribute Type

The following command deletes the attribute type that is created in Example 11–1:

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
^D
```

Notice that you must include the `X-ORIGIN 'user defined'`, which was added by Directory Server to classify this schema definition as an extension.

Managing Object Classes Over LDAP

This section explains how to create, view, and delete object classes over LDAP.

Creating Object Classes

The `cn=schema` entry has a multivalued attribute, `objectClasses`, that contains definitions of each object class in the directory schema. You can add to those definitions by using the `ldapmodify(1)` command.

New object class definitions, and changes that you make to user-defined object classes, are saved in the file `99user.ldif`.

If you are creating several object classes that inherit from other object classes, you must create the parent object classes first. If your new object class uses custom attributes, you must also define those first.

For each object class definition, you must provide at least an OID. Consider using at least the following elements for new object classes:

- **Object Class OID.** Corresponds to the object identifier for your object class. An OID is a string, usually of dotted decimal numbers, that uniquely identifies the schema object.

  For strict LDAP v3 compliance, you must provide a valid numeric OID. To learn more about OIDs or to request a prefix for your enterprise, send email to the IANA (Internet Assigned Number Authority) at `iana@iana.org`, or see the IANA web site (`http://www.iana.org`).
- **Object class name.** Corresponds to a unique name for the object class.
- **Parent object class.** Is an existing object class from which this object class inherits attributes.

If you do not intend to have this object class inherit from another specific object class, use `top`.

Typically, if you want to add new attributes for user entries, the parent would be the `inetOrgPerson` object class. If you want to add new attributes for corporate entries, the parent is usually `organization` or `organizationalUnit`. If you want to add new attributes for group entries, the parent is usually `groupOfNames` or `groupOfUniqueNames`.

- **Required attributes.** Lists and defines attributes that *must* be present for this object class.
- **Allowed attributes.** Lists and defines additional attributes that can be present for this object class.

▼ To Create an Object Class

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Prepare your object class definition according to the syntax specified in RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`).**

2. **Use the `ldapmodify`(1) command to add your object class definition.**

   Notice that Directory Server adds `X-ORIGIN 'user defined'` to the definition that you provide.

Example 11–4 Creating an Object Class

The following example adds a new object class using the `ldapmodify` command:

```
$ cat blogger.ldif
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 1.2.3.4.5.6.8
 NAME 'blogger'
 DESC 'Someone who has a blog'
 SUP inetOrgPerson
 STRUCTURAL
 MAY blog )

$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogger.ldif
Enter bind password:
modifying entry cn=schema

$
```

In a production environment, you would provide a valid, unique OID, not `1.2.3.4.5.6.8`.

Viewing Object Classes

The cn=schema entry has a multivalued attribute, objectClasses, that contains definitions of each object class in the directory schema. You can read those definitions by using the ldapsearch(1) command.

▼ To View an Object Class

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Use the ldapsearch command to view all the object class definitions that currently exist in your directory schema.**

Example 11–5 Viewing Object Classes

The following command displays definitions for all object classes:

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses
```

The -T option prevents the ldapsearch command from folding LDIF lines, so you can more easily work with the output using commands such as grep or sed. If you then pipe the output of this command through the grep command, you can view only the user-defined extensions to directory schema. For example:

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses | grep "user defined"
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger'
 DESC 'Someone who has a blog' STRUCTURAL MAY blog
 X-ORIGIN 'user defined' )
$
```

Deleting Object Classes

The cn=schema entry has a multivalued attribute, objectClasses, that contains definitions of each object class in the directory schema. You can delete definitions with X-ORIGIN 'user defined' by using the ldapmodify(1) command.

Because the schema is defined by the LDAP view in cn=schema, you can view and modify the schema online using the ldapsearch and ldapmodify utilities. However, you can delete only schema elements that have the value 'user defined' for the X-ORIGIN field. The server will not delete other definitions.

Changes that you make to user-defined elements are saved in the file 99user.ldif .

▼ To Delete an Object Class

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **View the definition of the object class to delete.**

   See "To View an Object Class" on page 315 for details.

2. **Use the `ldapmodify(1)` command to delete the object class definition as it appears in the schema.**

Example 11–6 Deleting an Object Class

The following command deletes the object class that was created in Example 11–4:

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: objectClasses
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger' DESC 'Someone who has a blog'
 STRUCTURAL MAY blog X-ORIGIN 'user defined' )
^D
```

Notice that you must include `X-ORIGIN 'user defined'`, which was added by Directory Server to classify this schema definition as an extension.

Replicating Directory Schema

Whenever you configure the replication of one or more suffixes between two servers, schema definitions are automatically replicated as well. The automatic replication of schema definitions ensures that all replicas have a complete, identical schema that defines all object classes and attributes that can be replicated to the consumers. Master servers therefore also contain the master schema.

However, schema replication is not instantaneous, even when you modify schema over LDAP. Schema replication is triggered either by updates to directory data or at the start of the first replication session after the schema is modified.

To enforce the schema on all replicas, you must at least enable schema checking on all masters. As the schema is checked on the master where the LDAP operation is performed, the schema does not need to be checked when updating the consumer. To improve performance, the replication mechanism bypasses schema checking on consumer replicas.

Note – Do not turn off schema checking on hubs and dedicated consumers. Schema checking has no performance impact on a consumer. Keep schema checking on to indicate that the replica contents conform to its schema.

Master servers replicate the schema automatically to their consumers during consumer initialization. Master servers also replicate the schema automatically any time the schema is modified through DSCC or through the command-line tools. By default, the entire schema is replicated. Any additional schema elements that do not yet exist on the consumer are created on the consumer and stored in the `99user.ldif` file.

For example, assume that a master server contains schema definitions in the `98mySchema.ldif` file when the server is started. Also assume that you then define replication agreements to other servers, either masters, hubs, or dedicated consumers. When you subsequently initialize the replicas from this master, the replicated schema contains the definitions from `98mySchema.ldif`, but the definitions are stored in `99user.ldif` on the replica servers.

After the schema has been replicated during consumer initialization, modifying the schema in `cn=schema` on the master also replicates the entire schema to the consumer. Therefore, any modifications to the master schema through the command-line utilities or through DSCC are replicated to the consumers. These modifications are stored in `99user.ldif` on the master, and by the same mechanism as described previously, the modifications are also stored in `99user.ldif` on the consumers.

Consider the following guidelines for maintaining consistent schema in a replicated environment:

- Do not modify the schema on a consumer server.

  Modifications to the schema on the consumer server can cause replication errors. This is because differences in the schema on the consumer can result in updates from the supplier not conforming to schema on the consumer.

- In a multimaster replication environment, modify schema on a single master server.

  When you modify the schema on two master servers, the master most recently updated propagates its version of the schema to the consumer. The schema on the consumer might then become inconsistent with the schema on the other master.

When configuring *fractional replication*, also consider the following guidelines:

- As schema is pushed by suppliers in fractional replication configurations, schema on a fractional consumer replica are a copy of the master replica's schema. Therefore, schema might not correspond to the fractional replication configuration being applied.
- In general, Directory Server replicates all required attributes for each entry as defined in the schema to avoid schema violations. When you configure fractional replication to filter out required attributes, you must disable schema checking.

- If schema checking is enabled with fractional replication, you might not be able to initialize the replica offline. Directory Server does not allow you to load data from LDIF if required attributes are filtered out.
- If you have disabled schema checking on a fractional consumer replica, schema checking is not enforced on the whole server instance on which that fractional consumer replica resides. As a result, avoid configuring supplier replicas on the same server instance as a fractional consumer.

Limiting Schema Replication

By default, whenever the replication mechanism replicates the schema, it sends the entire schema to its consumers. In the following situation, it is not desirable to send the entire schema to consumers:

Modifications to `cn=schema` using DSCC or from the command-line are limited to the user-defined schema elements, and all of the standard schema are unchanged. If you modify the schema often, sending the large set of unchanged schema elements every time has a performance impact. You might be able to improve replication and server performance by replicating only the user-defined schema elements.

▼ To Limit Schema Replication

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **Limit schema replication so that only the user-defined schema is replicated.**

  ```
  $ dsconf set-server-prop -h host -p port repl-user-schema-enabled:on
  ```

  The default value, `off`, causes the entire schema to be replicated when necessary.

◆ ◆ ◆ CHAPTER 12

Directory Server Indexing

Like a book index, Directory Server indexes speed up searches by associating search strings with references to the directory contents.

For information about the types of indexes and index tuning, see Chapter 9, "Directory Server Indexing," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Managing Indexes

This section describes how to manage indexes for specific attributes. The section includes information about creating, modifying, and deleting indexes. See "Managing Browsing Indexes" on page 333 for procedures specific to virtual list view (VLV) operations.

You can also check the attributes that need to be indexed by running the `dsconf info` command.

▼ To List Indexes

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **To list your existing indexes and their properties, use this command:**

```
$ dsconf list-indexes -h host -p port -v suffix-DN
```

▼ To Create Indexes

Note – You cannot create a new system index. Only the existing system indexes defined internally by Directory Server are maintained.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Create the new index configuration.**

Use the `dsconf create-index` command-line utility to configure the new index information by specifying the attribute that you want to index.

For example, to create an index entry for the `preferredLanguage` attribute, use this command:

```
$ dsconf create-index -h host -p port dc=example,dc=com preferredLanguage
```

Note – The command `dsconf create-index` sets the index configuration, but does not actually create the index files necessary for searches. Generating the index files can affect performance. To give you more control over the indexing procedure, generating the index files is done manually after the new index configuration has been created.

Always use the attribute's primary name when creating indexes. Do not use the attribute's alias. The primary name of the attribute is the first name listed for the attribute in the schema, for example, `uid` for the `userid` attribute.

2 **(Optional) Set the index properties by using the `dsconf set-index-prop` command.**

The `dsconf create-index` command creates an index with default properties. If you want to modify these properties, use the `dsconf set-index-prop` command. For more information about modifying index properties, see "To Modify Indexes" on page 321.

Note – When the configuration of an index is modified, re-indexing is required for the changes to take effect and to use the index again.

3 **Generate the index files.**

See "To Generate Indexes" on page 322.

4 **Repeat the previous steps for all servers that you want to be indexed.**

▼ To Modify Indexes

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Modify the index properties.

```
$ dsconf set-index-prop -h host -p port suffix-DN attr-name property:value
```

For example, to enable the approximate index `approx-enabled` for the `preferredLanguage` index , use the command:

```
$ dsconf set-index-prop -h host -p port dc=example,dc=com \
preferredLanguage approx-enabled:on
```

You can modify the following properties for each index:

- `eq-enabled` equality
- `pres-enabled` presence
- `sub-enabled` substring

One of the properties that you might want to modify is the optional `nsMatchingRule` attribute. This attribute contains the OID for any matching rule known by the server. It enables the OID of a language collation order for internationalized indexes, and other matching rules such as `CaseExactMatch`. For a list of supported locales and the OID of their associated collation order, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

For more information about index configuration attributes, see *Sun Directory Server Enterprise Edition 7.0 Reference*.

Note – When the configuration of an index is modified, re-indexing is required for the changes to take effect and to use the index again.

Run `dsconf info` to display the attributes that need to be reindexed. For example, the following output shows the `cn` and `uid` attributes that need to be reindexed.

```
$ dsconf info
Instance path    :  /local/dsInst
Global State     :  read-write
Host Name        :  hostname
Port             :  port
Secure port      :  secure-port
Total entries    :  160

Suffixes         :  dc=example,dc=com

No active tasks

dc=example,dc=com
=================
Attribute to reindex  :  cn
                         uid
```

2 **Regenerate your new indexes.**

See "To Generate Indexes" on page 322.

3 **Repeat the previous steps for all servers that include the modified attribute index.**

▼ To Generate Indexes

This procedure generates index files so that new or modified indexes can be searchable. If you modify an index configuration for an attribute, all searches that include that attribute as a filter are not indexed. To ensure that searches including that attribute are successful, use this procedure command to regenerate existing indexes every time you create or modify an index configuration for an attribute.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Generate your index files in one of the following ways:**
  - Generate your new index files online.

    ```
    $ dsconf reindex -h host -p port [-t attr] suffix-DN
    ```

where `-t` specifies that only the specified attribute or attributes are to be reindexed, not all attributes.

For example, to regenerate the `preferredLanguage` index, type:

```
$ dsconf reindex -h host -p port -t preferredLanguage dc=example,dc=com
```

While the `dsconf reindex` command is running, the contents of the suffix remain available through the server. However, searches are not indexed until the command has completed. Reindexing is a resource-intensive task that can impact the performance of other operations on the server.

- Generate your new index files offline.

  ```
  $ dsadm reindex -t attr instance-path suffix-DN
  ```

  For example, to regenerate the `preferredLanguage` index, type:

  ```
  $ dsadm reindex -t preferredLanguage /local/dsInst dc=example,dc=com
  ```

- Regenerate all of your indexes quickly offline by reinitializing your suffix.

  When you reinitialize a suffix, all index files are automatically regenerated. Depending on the size of the directory, reinitializing the suffix is usually faster than reindexing two or more attributes. However, the suffix is unavailable during the initialization. For more information, see "Reindexing a Suffix by Reinitialization" on page 332.

Analyzing Indexes

Analyzing indexes requires gathering search filter usage patterns from user requests, especially for values as specified in the filters and subsequently looking the status of those values into the index files. When a search requests is processed, internal or user generated, a first phase uses indexes to find potential entries to be returned. Depending on the search filter, if the index of one of the specified attributes matches too many entries (ALLIDs), the search becomes non-indexed (`notes=U` in the access log). In any case, all entries thus gathered (or the entire DIT) are matched against the filter for actually returning the search result.

Note – Directory Server tries to optimize the search so that not all index values matching `ALLID` will force the search to be not indexed.

▼ To Analyze Index Filters

The `dsconf analyze-index-filters` command when enabled, displays the Directory Server search filters with their statistics.

1 Enable index filter analyzer using the following command:

```
$ dsconf enable-index-filter-analyzer [--max-entries INT] SUFFIX_DN
```

For example, to enable index filter analyzer on `o=example.com`, run the following command:

```
$ dsconf enable-index-filter-analyzer -p port-number o=example.com
$ dsconf get-suffix-prop -p port-number o=example.com \
index-filter-analyzer-enabled index-filter-analyzer-max-entries


index-filter-analyzer-enabled      :  on
index-filter-analyzer-max-entries  :  2000
```

2 Analyze index filters to know the indexes that need to be reindexed.

```
$ dsconf analyze-index-filters -p port-number SUFFIX_DN
```

The output displays all the attributes that need to be reindexed, index filter usage statistics, number of hits, and number of `allids` hits. Based on the output, you can configure indexes differently by modifying the `all-ids-threshold` property or by creating indexes, to improve performance.

```
$ dsconf analyze-index-filters -p port-number o=example.com

Observations started at Nov 13, 2008 12:01:29 PM

Total number of search requests :   8
Total number of Allids :            7

filter                Type  #allids/#hits  Threshold  Max matching  Additional info
                                                      entries
-------------------   ----  -------------  ---------  ------------  ---------------
(departmentNumber       eq            1/1                       2  "departmentNumber"
 =9415)                                                            is not indexed

(objectClass            eq            1/1      *4000         10000  To investigate
 =inetOrgPerson)

(objectClass=*)       pres            1/1                   10006  "pres" type is
                                                                    disabled for
                                                                    "objectClass"
                                                                    system index

(roomNumber=*)        pres            1/1                   10000  "roomNumber" is
                                                                    not indexed

(roomNumber=1*)         eq            1/1                    4071  "roomNumber" is
                                                                    not indexed

(telephoneNumber=*)   pres            2/2      *4000         10000  To investigate

(telephoneNumber=1*)    eq            0/1       4000         10000
```

```
## * indicates thresholds which have been crossed.

No attributes need to be reindexed

Use "dsconf set-index-prop o=example.com ATTR_NAME..." to set the allids threshold
value and to take benefit of indexes.
```

The displayed filters are the basic filter elements along with the following information:

- Type that is used during processing.
- How many times the filter element is used and how many times the index value is `Allids`.

Note – `Allids` can occur because of unsufficient privileges or if the index type configuration is changed. In the latter case, data may be skewed.

Complex filters are broken down to these basic elements and will not appear in their entirety.

The example output displays the following information:

- Some index values are `Allids`
- Some attributes are not indexed, that is, `roomNumber`, `departmentNumber`.
- Total `4071` entries matched the `roomNumber=1*` filter
- Some indexes are well configured for some filter use, that is, `telephoneNumber=1*`.

3 Create the new index for the attribute `roomNumber`.

```
$ dsconf create-index -p port-number o=example.com roomNumber
```

4 Run the `analyze-index-filters` command again to check the status.

```
$ dsconf analyze-index-filters -p port-number o=example.com

Observations started at Nov 13, 2008 12:01:29 PM

Total number of search requests :   9
Total number of Allids :            8

filter               Type  #allids/#hits  Threshold  Max matching  Additional info
                                                     entries
-------------------  ----  -------------  ---------  ------------  -----------------
(departmentNumber       eq            1/1                      2  "departmentNumber"
 =9415)                                                            is not indexed

(objectClass            eq            1/1      *4000         10000  To investigate
=inetOrgPerson)
```

```
(objectClass=*)       pres            1/1                   10006  "pres" type is
                                                                     disabled for
                                                                     "objectClass"
                                                                     system index

(roomNumber=*)        pres            1/1      *4000        10000  To investigate

(roomNumber=1*)         eq            1/1      *4000         4071  To investigate

(telephoneNumber=*)   pres            2/2      *4000        10000  To investigate

(telephoneNumber=1*)    eq            0/1       4000        10000

## * indicates thresholds which have been crossed.

Attributes to reindex :  roomNumber

Use "dsconf reindex --attr ATTR_NAME... o=example.com" to reindex.
Use "dsconf set-index-prop o=example.com ATTR_NAME..." to set the allids
threshold value and to take benefit of indexes.
```

As required, follow the appropriate procedure as displayed at the end of the output.

5 Restart the index filter analyzer to consider the latest index updates.

To restart the index filter analyzer, disable the analyzer using the `disable-index-filter-analyzer` subcommand and then start the analyzer again using the `enable-index-filter-analyzer` subcommand.

Note – Monitoring affects performance. It also requires heavy memory resources, based on the configured maximum number of filters to monitor.

Run the `dsconf info` command to know when the analyzer was enabled. If you do not want to monitor indexes and analyze-index-filters, it is not recommended to keep the analyzer running.

Directory Server tries to optimize the search so that when complex filters are evaluated, not all elements could be processed. Do not expect a one-to-one relation with what appears in the access log, and a complex filter and its constituent elements.

See Also For more information, see `dsconf(1M)`.

▼ To Analyze Attribute Indexes

Using the `dsconf analyze-index-filters` command, gather the most used filters and their behavior. On the other hand, to know the data as appears in the index files, use `dsadm analyze-indexes` to have a snapshot of index files.

1 **Your server must be stopped before analyzing the attribute indexes.**

```
$ dsadm stop INSTANCE_PATH
```

2 **Analyze attribute indexes using the following command:**

```
$ dsadm analyze-indexes [-bRi] [-o FILE] INSTANCE_PATH SUFFIX_DN
```

For example, to analyze the attribute indexes of suffix `o=example.com`, run the following command:

```
$ dsadm analyze-indexes /local/myinst o=example.com
This operation may take a long time and generate important amounts of data
Do you want to continue [y/n]? y
Generating raw index data, please wait...
Raw index data available in file '/local/myinst/logs/db_stat_example%2ecom'

Index            Type       Total Keys  ALLIDs  95%  90%  80%
---------------  ---------  ----------  ------  ---  ---  ---
aci              PRESENCE            1       0    0    0    0
ancestorid       EQUALITY            6       6    0    0    0
cn               EQUALITY       200000       0    0    0    0
cn               SUBSTRING       14828      15    0    0    0
entrydn          EQUALITY       100006       0    0    0    0
givenName        EQUALITY         8605       0    0    0    0
givenName        SUBSTRING        4762       4    0    0    0
givenName        PRESENCE            1       1    0    0    0
mail             EQUALITY       100000       0    0    0    0
mail             SUBSTRING       14975      26    1    3    2
mail             PRESENCE            1       1    0    0    0
nsuniqueid       EQUALITY       100007       0    0    0    0
numsubordinates  PRESENCE            1       0    0    0    0
objectclass      EQUALITY            7       4    0    0    0
parentid         EQUALITY            6       5    0    0    0
sn               EQUALITY       100000       0    0    0    0
sn               SUBSTRING       12993       0    0    0    0
telephoneNumber  EQUALITY        99924       0    0    0    0
telephoneNumber  SUBSTRING        1106      24    0    0    0
telephoneNumber  PRESENCE            1       1    0    0    0
uid              EQUALITY       200000       0    0    0    0
uid              PRESENCE            1       1    0    0    0

aci PRESENCE
============
ALLIDs keys : 0 / 1

ancestorid EQUALITY
===================
ALLIDs keys : 6 / 6
     [1] [2] [3] [4] [5] [6]
```

```
cn EQUALITY
===========
ALLIDs keys : 0 / 200000

cn SUBSTRING
============
ALLIDs keys : 15 / 14828
     [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [^us]
     [er1] [r10] [ser] [use]

entrydn EQUALITY
================
ALLIDs keys : 0 / 100006

givenName EQUALITY
==================
ALLIDs keys : 0 / 8605

givenName SUBSTRING
===================
ALLIDs keys : 4 / 4762
     [^ma] [ie$] [na$] [ne$]

givenName PRESENCE
==================
ALLIDs keys : 1 / 1
    [pres]

mail EQUALITY
=============
ALLIDs keys : 0 / 100000

mail SUBSTRING
==============
ALLIDs keys : 26 / 14975
     [.co] [0@e] [1@e] [2@e] [3@e] [4@e] [5@e] [6@e] [7@e] [8@e] [9@e]
     [@ex] [^ma] [amp] [com] [ell] [exa]  [e.c] [ie_] [le.] [mar] [mpl]
     [na_] [ne_] [om$] [ple] [xam]

mail PRESENCE
=============
ALLIDs keys : 1 / 1
    [pres]

nsuniqueid EQUALITY
===================
ALLIDs keys : 0 / 100007
```

```
numsubordinates PRESENCE
========================
ALLIDs keys : 0 / 1

objectclass EQUALITY
====================
ALLIDs keys : 4 / 7
     [inetorgperson] [organizationalperson] [person] [top]

parentid EQUALITY
=================
ALLIDs keys : 5 / 6
     [2] [3] [4] [5] [6]

sn EQUALITY
===========
ALLIDs keys : 0 / 100000

sn SUBSTRING
============
ALLIDs keys : 0 / 12993

telephoneNumber EQUALITY
========================
ALLIDs keys : 0 / 99924

telephoneNumber SUBSTRING
=========================
ALLIDs keys : 24 / 1106
     [120] [121] [130] [140] [141] [151] [171] [180] [181] [206] [213]
     [303] [408] [415] [510] [714] [804] [818] [^12] [^13] [^14] [^15]
     [^17] [^18]

telephoneNumber PRESENCE
========================
ALLIDs keys : 1 / 1
    [pres]

uid EQUALITY
============
ALLIDs keys : 0 / 200000

uid PRESENCE
============
ALLIDs keys : 1 / 1
    [pres]
```

Following what explained in Chapter 9, "Directory Server Indexing," in *Sun Directory Server Enterprise Edition 7.0 Reference*, `dsadm analyze-indexes` displays the status of the value keys as used by Directory Server. If most of the keys are `Allid` or 95% of `Allid`, the number of entries matching the key is at least equal to `all-ids-threshold`. The index most likely has to be configured with a higher value for `all-ids-threshold`.

Note – Too high an `all-ids-threshold` value can impact performance.

The `dsadm analyze-indexes` displays which keys are ALLID or close to be, so it can be matched with the output of `dsconf analyze-index-filters`. If a search specifies a filter whose value is an allid key, the search might not be indexed, depending on the entire search filter as mentioned above.

▼ To Delete Indexes

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Remove all indexes that are configured for an attribute.**

  ```
  $ dsconf delete-index -h host -p port suffix-DN attr-name
  ```

  For example, the following command deletes all indexes for the `preferredLanguage` attribute:

  ```
  $ dsconf delete-index -h host -p port dc=example,dc=com preferredLanguage
  ```

  Take great care when deleting default indexes because it can affect Directory Server functioning.

Changing the Index List Threshold

Slow searches might be a result of your system index list size exceeding the index list threshold. The index list threshold is the maximum number of values for each index key. To determine whether the index list threshold size has been exceeded, examine the access log. The `notes=U` flag at the end of an access log `RESULT` message indicates that an unindexed search was performed. A previous `SRCH` message for the same connection and operation specifies the search filter that was used. The following two-line example traces an unindexed search for `cn=Smith` that returns 10,000 entries. Timestamps have been removed from the messages.

```
conn=2 op=1 SRCH base="o=example.com" scope=0 filter="(cn=Smith)"
conn=2 op=1 RESULT err=0 tag=101 nentries=10000 notes=U
```

If your system often exceeds the index list threshold, consider increasing the threshold to improve performance. The following procedure uses the `dsconf set-server-prop` command to modify the `all-ids-threshold` property.

▼ To Change the Index List Threshold

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Adjust the index list threshold.

You can adjust your index list threshold at any of the following levels:

- At the instance level:

  ```
  dsconf set-server-prop -h host -p port all-ids-threshold:value
  ```

- At the suffix level:

  ```
  dsconf set-suffix-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- At the entry level:

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- At the index level, by search type:

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold search-type:value
  ```

  where *search-type* is one of the following:

  - `eq-enabled` equality
  - `pres-enabled` presence
  - `sub-enabled` substring

  The `all-ids-threshold` property cannot be configured for the approximate index.

You can use DSCC to set the threshold at the index level, by search type. For more information, see the Directory Server online help.

2 Regenerate the suffix indexes.

See "To Generate Indexes" on page 322.

3 If the database cache size was tuned for the old `all IDs` threshold value and the server has adequate physical memory, consider increasing the database cache size.

Increase the database cache size by 25 percent of the magnitude of the increase to the `all IDs` threshold.

In other words, if you increase the `all IDs` threshold from 4000 to 6000, you can increase the database cache size by about 12 ½ percent to account for the increase in index list size.

Database cache size is set using the attribute `dbcachesize`. Find the optimum size empirically before applying changes to production servers.

Reindexing a Suffix

If your index files become corrupt, or if you change the index for an attribute, you must reindex the suffix to recreate the index files in the corresponding database directory. You can reindex a suffix while the directory server is running or by reinitializing the suffix.

Reindexing a Suffix While the Directory Server Is Running

When you reindex a suffix, the server examines all of the entries the suffix contains and rebuilds the index files. During reindexing, the contents of the suffix are read-only. Because the server must scan the entire suffix for every attribute that is reindexed, this process might take up to several hours for suffixes with millions of entries. The length of time also depends on the indexes you configure. In addition, while the suffix is being reindexed, it is not available.

▼ To Reindex All Indexes on A Suffix

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Reindex all indexes on a suffix.**

  ```
  $ dsconf reindex -h host -p port suffix-DN
  ```

  For example, to initialize all indexes on the `dc=example,dc=com` suffix, use this command:

  ```
  $ dsconf reindex -h host -p port dc=example,dc=com
  ```

Reindexing a Suffix by Reinitialization

When you reinitialize a suffix, the new contents are imported, which means that the suffix contents are replaced and new index files are created.

▼ To Reindex a Suffix Through Reinitialization

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Set the suffix to read-only, as described in "Setting Referrals and Making a Suffix Read-Only" on page 50.**

2. **Export the entire suffix to an LDIF file, as described in "Backing Up to LDIF" on page 225.**

3. **Import the same LDIF file to reinitialize the suffix, as described in "Importing Data From an LDIF File" on page 52.**

   During the initialization, the suffix is unavailable. When the initialization is complete, all configured indexes are ready to be used.

4 **Make the suffix writable again, as described in "Setting Referrals and Making a Suffix Read-Only" on page 50.**

Managing Browsing Indexes

Browsing indexes are special indexes used only for search operations that request server-side sorting of results. *Sun Directory Server Enterprise Edition 7.0 Reference* explains how browsing indexes work in Directory Server.

Browsing Indexes for Client Searches

Customized browsing indexes for sorting client search results must be defined manually. To create a browsing index, or virtual list view (VLV) index, use the following procedure. This section also includes procedures for adding or modifying browsing index entries and for regenerating browsing indexes.

▼ To Create a Browsing Index

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1 **Add new browsing index entries or edit existing browsing index entries by using the `ldapmodify` command.**

For instructions, see "To Add or Modify Browsing Index Entries" on page 333.

2 **Run the `dsconf reindex` command to generate the new set of browsing indexes to be maintained by the server.**

For instructions, see "To Regenerate Browsing Indexes" on page 335.

▼ To Add or Modify Browsing Index Entries

A browsing index is specific to a given search on a given base entry and its subtree. The browsing index configuration is defined in the database configuration of the suffix that contains the entry.

1 **Configure the `vlvBase`, `vlvScope`, and `vlvFilter` attributes for each browsing index on a directory server.**

These attributes configure the base of the search, the scope of the search, and a filter for the search. These attributes use the `vlvSearch` object class.

2 Configure the `vlvSort` attribute for each browsing index.

This attribute specifies the name of the attribute or attributes that sort the index. This entry is a child of the first entry and uses the `vlvIndex` object class to specify which attributes to sort and in what order.

The following example uses the `ldapmodify` command to create the browsing index configuration entries:

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=people_browsing_index, cn=database-name,
cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvSearch
cn: Browsing ou=People
vlvBase: ou=People,dc=example,dc=com
vlvScope: 1
vlvFilter: (objectclass=inetOrgPerson)

dn: cn=Sort rev employeenumber, cn=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort rev employeenumber
vlvSort: -employeenumber
^D
```

The `vlvScope` is one of the following:

- `0` for the base entry alone
- `1` for the immediate children of the base
- `2` for the entire subtree rooted at the base

The `vlvFilter` is the same LDAP filter that is used in the client search operations. Because all browsing index entries are located in the same place, you should use descriptive `cn` values to name your browsing indexes.

Each `vlvSearch` entry must have at least one `vlvIndex` entry. The `vlvSort` attribute is a list of attribute names that defines the attribute to sort on and the sorting order. The dash (`-`) in front of an attribute name indicates reverse ordering. You can define more than one index for a search by defining several `vlvIndex` entries. With the previous example, you could add the following entry:

```
$ ldapmodify -a -h host -p port
 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Sort sn givenname uid, cn=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
```

```
cn: Sort sn givenname uid
vlvSort: sn givenname uid
^D
```

3. **To modify a browsing index configuration, edit the corresponding `vlvSearch` entry or the corresponding `vlvIndex` entry.**

4. **To remove a browsing index so that the browsing index is no longer maintained by the server, remove the individual `vlvIndex` entries.**

   Alternatively, if only one `vlvIndex` entry exists, remove both the `vlvSearch` entry and the `vlvIndex` entry.

▼ To Regenerate Browsing Indexes

- **After you have created the browsing index entries, generate the new browsing indexes for the attributes specified.**

  ```
  $ dsadm reindex -l -t attr-index instance-path suffix-DN
  ```

  The command scans the directory contents and creates a database file for the browsing index.

  The following example generates the browsing index that you defined in the previous section:

  ```
  $ dsadm reindex -l -b database-name -t Browsing /local/dsInst \
   ou=People,dc=example,dc=com
  ```

  For more information about the `dsadm reindex` command, see the `dsadm(1M)` man page.

◆ ◆ ◆ CHAPTER 13

Directory Server Attribute Value Uniqueness

The UID uniqueness plug-in ensures that the value of a given attribute is unique among all entries of the directory or of a subtree. The plug-in stops any operation that tries to add an entry that contains an existing value for the given attribute. The plug-in also stops any operation that adds or modifies the attribute to a value that already exists in the directory.

The UID uniqueness plug-in is disabled by default. When the plug-in is enabled, it ensures the uniqueness of the `uid` attribute by default. You can create new instances of the plug-in to enforce unique values on other attributes. The UID uniqueness plug-in ensures attribute value uniqueness on a single server.

This chapter covers the following topics:

Overview of Attribute Value Uniqueness

The UID uniqueness plug-in is a pre-operation plug-in. It checks LDAP add, modify, and modify DN operations before the server performs an update of the directory. The plug-in determines whether the operation will cause two entries to have the same attribute value. If so, the server terminates the operation and returns error 19 `LDAP_CONSTRAINT_VIOLATION` to the client.

You can configure the plug-in to enforce uniqueness in one or more subtrees in the directory or among entries of a specific object class. This configuration determines the set of entries for which unique attribute values is enforced.

You can define several instances of the UID uniqueness plug-in if you want to enforce the uniqueness of other attributes. Define one plug-in instance for each attribute whose value must

be unique. You can also have several plug-in instances for the same attribute to enforce "separate" uniqueness in several sets of entries. A given attribute value is allowed only once in each set of subtrees.

When you enable attribute uniqueness on an existing directory, the server does not check for uniqueness among existing entries. Uniqueness is only enforced when an entry is added or when the attribute is added or modified.

By default, the UID uniqueness plug-in is disabled because the plug-in affects multimaster replication. You can enable the UID uniqueness plug-in when using replication, but you should be aware of the behavior described in "Using the Uniqueness Plug-In With Replication" on page 340.

Enforcing Uniqueness of the `uid` and Other Attributes

This section explains how to enable and configure the default uniqueness plug-in for the `uid` attribute and how to enforce uniqueness of any other attribute.

▼ To Enforce Uniqueness of the `uid` Attribute

This procedure describes how to enable and configure the UID uniqueness plug-in by using the `dsconf` command. The DN of the plug-in configuration entry is `cn=uid uniqueness,cn=plugins,cn=config`.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

When using DSCC, you must not modify the default UID uniqueness plug-in to enforce uniqueness of another attribute. If you do not want to have a UID uniqueness plug-in, leave the plug-in disabled and create a new plug-in instance for another attribute, as described in "To Enforce Uniqueness of Another Attribute" on page 339.

1. **Enable the plug-in.**

   ```
   $ dsconf enable-plugin -h host -p port "uid uniqueness"
   ```

2. **Modify the plug-in arguments according to how you want to specify the subtrees where uniqueness is enforced.**

   - To specify the base DN of a single subtree, type:

     ```
     $ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid\
      argument:subtreeBaseDN
     ```

     For example:

```
$ dsconf set-plugin-prop -h host1 -p 1389 "uid uniqueness" argument:uid \
 argument:dc=People,dc=example,dc=com
```

- To specify more than one subtree, add more arguments with the full base DN of a subtree as the value of each argument.

  ```
  $ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid \
   argument:subtreeBaseDN argument:subtreeBaseDN
  ```

- To specify subtrees according to the object class of their base entries, set the arguments to the following values. Uniqueness of the uid attribute is enforced in the subtree below every entry with the *baseObjectClass*. Optionally, you can specify the *entryObjectClass* in the third argument so that the plug-in only enforces uniqueness in operations that target entries with this object class.

  ```
  $ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:attribute=uid \
   argument:markerObjectClass=baseObjectClass argument:entryObjectClass=baseObjectClass
  ```

- To add an argument to an existing list of arguments, use the following command:

  ```
  $ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument+:argument-value
  ```

3 Restart the server for your changes to take effect.

▼ To Enforce Uniqueness of Another Attribute

The UID uniqueness plug-in can be used to enforce the uniqueness of any attribute. You must create a new instance of the plug-in by creating a new entry under cn=plugins,cn=config in the directory.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Create a new plug-in.

```
$ dsconf create-plugin -h host -p port -H lib-path -F init-func \
 -Y type plugin-name
```

plugin-name should be a short and descriptive name that includes the name of the attribute. For example, to create a plug-in for the uniqueness of the mail ID attribute, use this command:

```
$ dsconf create-plugin -h host1 -p 1389 -H /opt/SUNWdsee7/lib/sparcv9/uid-plugin.so \
 -F NSUniqueAttr_Init -Y preoperation "mail uniqueness"
```

2 Set the plug-in properties.

```
$ dsconf set-plugin-prop -h host -p port plugin-name property:value
```

For example, to set the properties for the mail uniqueness plug-in, :

```
$ dsconf set-plugin-prop -h host1 -p 1389 "mail uniqueness" \
 desc:"Enforce unique attribute values..." version:6.0 \
 vendor:"Sun Microsystems, Inc." depends-on-type:database
```

3 Enable the plug-in.

```
$ dsconf enable-plugin -h host -p port plugin-name
```

4 Specify the plug-in arguments.

These arguments depend on how you want to determine the subtrees where uniqueness is enforced.

- To define one or more subtrees according to their base DN, the first argument must be the name of the attribute that should have unique values. Subsequent arguments are the full DNs of the base entries of the subtrees.

  ```
  $ dsconf set-plugin-prop -h host -p port plugin-name argument:attribute-name \
   argument:subtreeBaseDN argument:subtreeBaseDN...
  ```

- To add an argument to an existing list of arguments, use the following command:

  ```
  $ dsconf set-plugin-prop -h host -p port plugin-name argument+:argument-value
  ```

- To define subtrees according to the object class of their base entries, the first argument must contain `attribute=`*attribute-name* which specifies the name of the attribute that should have unique values. The second argument must be the *baseObjectClass* that determines the base entry of subtrees where uniqueness is enforced. Optionally, you can specify an *entryObjectClass* in the third argument so that the plug-in enforces uniqueness only in operations that target entries with this object class.

  ```
  $ dsconf set-plugin-prop -h host -p port plugin-name\
   argument:attribute=attribute-name  argument:markerObjectClass=baseObjectClass\
   argument:requiredObjectClass=entryObjectClass
  ```

In all plug-in arguments, no space can appear before or after the = sign.

5 Restart the server for your changes to take effect.

Using the Uniqueness Plug-In With Replication

The UID uniqueness plug-in does not perform any checking on attribute values when an update is performed as part of a replication operation. This does not affect single-master replication, but the plug-in cannot automatically enforce attribute uniqueness for multimaster replication.

Single-Master Replication Scenario

Because all modifications by client applications are performed on the master replica, the UID uniqueness plug-in should be enabled on the master server. The plug-in should be configured to enforce uniqueness in the replicated suffix. Because the master ensures that the values of the desired attribute are unique, you do not need to enable the plug-in on the consumer server.

Enabling the UID uniqueness plug-in on the consumer of a single master does not interfere with replication or normal server operations. However, it might cause slight performance degradation.

Multimaster Replication Scenario

The UID uniqueness plug-in was not designed for use in a multimaster replication scenario. Because multimaster replication uses a loosely consistent replication model, simultaneously adding the same attribute value on both servers will not be detected, even if the plug-in is enabled on both servers.

However, you can use the UID uniqueness plug-in if the attribute on which you are performing the uniqueness check is a naming attribute, and the uniqueness plug-in is enabled for the same attribute in the same subtrees on all masters.

When these conditions are met, uniqueness conflicts are reported as naming conflicts at replication time. Naming conflicts must be resolved manually. For more information, refer to "Solving Common Replication Conflicts" on page 294.

◆ ◆ ◆ CHAPTER 14

Directory Server Logging

This chapter describes how to manage Directory Server logs.

If you want information to assist you in defining a logging strategy, use the logging policy information in "Designing a Logging Strategy" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

For a description of the log files and their contents, see Chapter 10, "Directory Server Logging," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Log Analysis Tool

The Directory Server Resource Kit provides a log analysis tool, `logconv`, that enables you to analyze Directory Server access logs. The log analysis tool extracts usage statistics. It also counts the occurrences of significant events. For more information about this tool, see the `logconv(1)` man page.

Viewing Directory Server Logs

You can view the logs directly on the server in the default *instance-path*/`logs` file. If you have modified the default path, you can find the log file location using the `dsconf` command as follows:

```
$ dsconf get-log-prop -h host -p port log-type path
```

Alternatively, you can view the log files through Directory Service Control Center (DSCC). DSCC enables you to view and sort the log entries.

The following figure shows a sample of a Directory Server access log in DSCC.

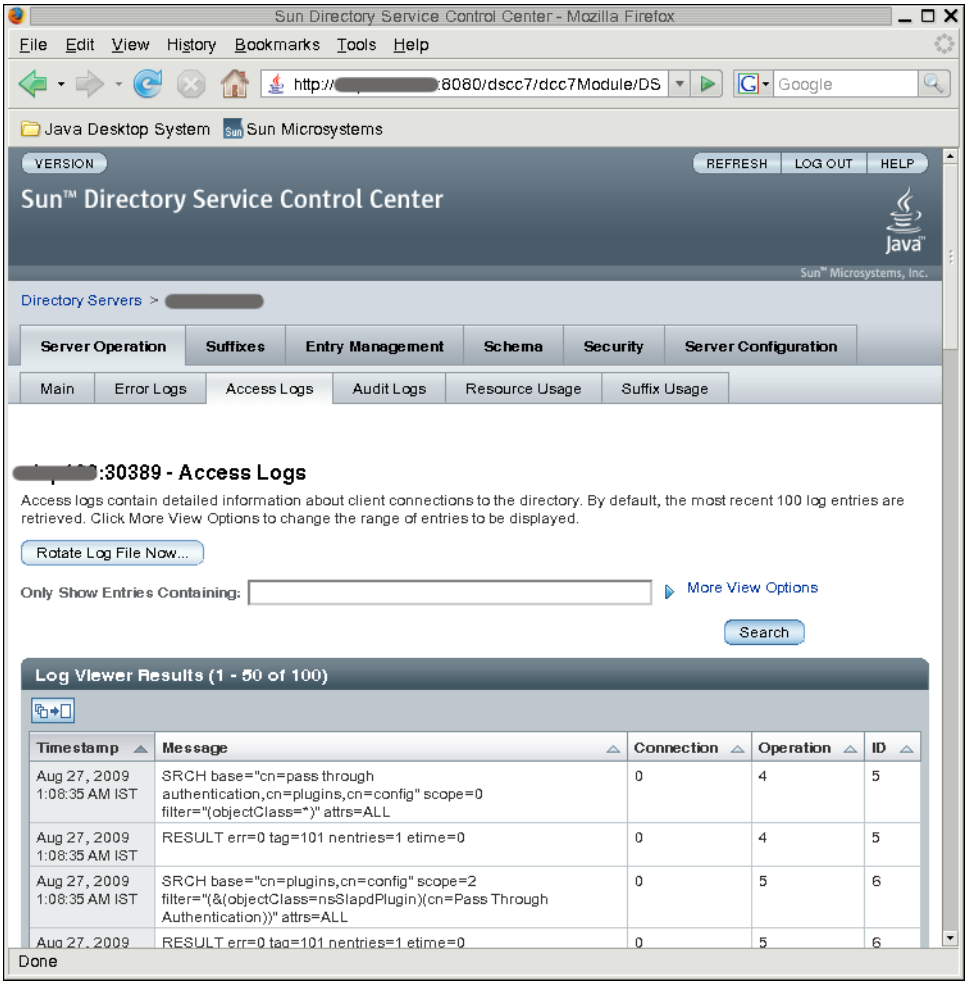


FIGURE 14–1 DSCC Access Log

▼ To Tail Directory Server Logs

You can use the `dsadm` command to display a specified number of lines of the Directory Server logs, or to display log entries younger than a specified age. This example tails the error log. To tail the access log, use `show-access-log` instead of `show-error-log`.

1 Display error log entries younger than a certain age.

```
$ dsadm show-error-log -A duration instance-path
```

You must specify a unit for the duration. For example, to display error log entries younger than 24 hours, type:

```
$ dsadm show-error-log -A 24h /local/dsInst
```

2 **Display a specified number of lines from the error log (starting from the end).**

```
$ dsadm show-error-log -L last-lines instance-path
```

The number of lines is expressed as an integer. For example, to display the last 100 lines, type:

```
$ dsadm show-error-log -L 100 /local/dsInst
```

If you do not specify a value, the default number of lines displayed is 20.

Configuring Logs for Directory Server

Many aspects of the log files can be modified. Some examples include the following:

- Enabling the audit log

  Unlike the access log and the errors log, the audit log is not enabled by default. For information, see "To Enable the Audit Log" on page 347.
- General settings
  - Enabling or disabling logging
  - Enabling or disabling log buffering
  - Log file location
  - Verbose logging
  - Log level
- Log rotation settings.
  - Creation of new logs at regular time intervals
  - Maximum log file size before a new log file is created
- Log deletion settings
  - Maximum file age before deletion
  - Maximum file size before deletion
  - Minimum free disk space before deletion

The following procedures describe how to modify log configuration and how to enable the audit log.

▼ To Modify Log Configuration

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **View the settings for the log that you want to modify.**

```
$ dsconf get-log-prop -h host -p port log-type
```

For example, to list the existing error log settings, type:

```
$ dsconf get-log-prop -h host1 -p 1389 error
Enter "cn=Directory Manager" password:
buffering-enabled         :  off
enabled                   :  on
level                     :  default
max-age                   :  1M
max-disk-space-size       :  100M
max-file-count            :  2
max-size                  :  100M
min-free-disk-space-size  :  5M
path                      :  /tmp/ds1/logs/errors
perm                      :  600
rotation-interval         :  1w
rotation-min-file-size    :  unlimited
rotation-time             :  undefined
verbose-enabled           :  off
```

2 **Set the new value.**

Set the value that you want for the property.

```
$ dsconf set-log-prop -h host -p port log-type property:value
```

For example, to set the rotation interval for the error log to two days, use this command:

```
$ dsconf set-log-prop -h host1 -p 1389 error rotation-interval:2d
```

▼ To Enable the Audit Log

Unlike the access log and errors log, the audit log is not enabled by default. Before viewing the audit log, you must enable it.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Enable the audit log.**

```
$ dsconf set-log-prop -h host -p port audit enabled:on
```

Rotating Directory Server Logs Manually

If you have a log that is getting very large, you can manually rotate the log at any time. Rotation backs up the existing log file and creates a fresh log file.

▼ To Rotate Log Files Manually

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Rotate the log file.**

  ```
  $ dsconf rotate-log-now -h host -p port log-type
  ```

  For example, to rotate the access log:

  ```
  $ dsconf rotate-log-now -h host1 -p 1389 access
  ```

◆ ◆ ◆ CHAPTER 15

Directory Server Monitoring

Directory Server can be monitored using a variety of methods. These methods are described in Chapter 6, "Directory Server Monitoring," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter describes how to set up and administer monitoring in Directory Server.

This chapter covers the following topics:

Setting Up SNMP for Directory Server

This section describes how to set up your server to be monitored through SNMP.

For a description of SNMP implementation in Directory Server, see "Directory Server and SNMP" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

▼ To Set Up SNMP

For parts of this procedure, you can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. Other parts of the procedure can only be done using the command line.

1 Enable the Java ES Monitoring Framework plug-in.

Use the procedure "Enabling Java ES MF Monitoring" on page 350. This procedure also enables the Common Agent Container, which is part of the Java ES MF.

2 **Access the SNMP-managed objects defined by the MIB and exposed through the agents.**

The tasks required for this step are entirely dependent on your SNMP management system. See your SNMP management system documentation for instructions.

When exposing the MIB, you might want to use the RFC text files for this MIB. These files are available at `http://www.ietf.org/rfc/rfc2605.txt`and `http://www.ietf.org/rfc/rfc2788.txt`.

Enabling Java ES MF Monitoring

If you want to use the Sun Java ES Monitoring Framework (Java ES MF) for monitoring, you must enable the Java ES MF plug-in.

For more information about administering the Java ES MF, see the *Sun Java Enterprise System 5 Update 1 Monitoring Guide*.

▼ To Enable Java ES MF Monitoring

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Initialize and register the Java ES Monitoring Framework.**

```
$ dsccsetup mfwk-reg
```

For the location of this command, see "Command Locations" on page 26.

2 **Enable the Java ES Monitoring Framework plug-in.**

```
$ dsconf enable-plugin -h host -p port "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Directory Server must be restarted for changes to take effect.
```

3 **Restart the Directory Server instance.**

```
$ dsadm restart instance-path
```

4 **Verify that the Java ES Monitoring Framework plug-in is enabled.**

```
$ dsconf get-plugin-prop -h host -p port -v "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Reading property values of the plugin "Monitoring Plugin"...
argument          :
depends-on-named  :
depends-on-type   :  database
desc              :  Monitoring plugin
enabled           :  on
```

```
feature           :  Monitoring
init-func         :  mf_init
lib-path          :  /opt/SUNWdsee7/lib/sparcv9/mf-plugin.so
type              :  object
vendor            :  Sun Microsystems, Inc.
version           :  7.0
```

Troubleshooting Java ES MF Monitoring

If Java ES MF monitoring does not work, ensure that you have correctly installed the Common Agent Container, as described in Chapter 2, "Installing Directory Server Enterprise Edition," in *Sun Directory Server Enterprise Edition 7.0 Installation Guide*.

If you are still experiencing problems, see the *Sun Java Enterprise System 5 Update 1 Monitoring Guide*.

Monitoring a Server Using `cn=monitor`

The following example shows how to view the general server statistics:

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -s base -b "cn=monitor" "(objectclass=*)"
```

For a description of all monitoring attributes that are available in these entries, see "Directory Server Monitoring Attributes" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

Many of the parameters that can be monitored reflect Directory Server performance, and are influenced by configuration and tuning. For more information about each of the configurable attributes, see the attribute man pages in *Sun Directory Server Enterprise Edition 7.0 Man Page Reference*.

PART II

Directory Proxy Server Administration

◆ ◆ ◆ CHAPTER 16

Directory Proxy Server Tools

Directory Proxy Server provides a browser interface and command-line tools to register and manage instances of Directory Proxy Server. The browser interface is called Directory Service Control Center (DSCC). This chapter describes basic tasks that are required to administer Directory Proxy Server by using DSCC or the command line.

To decide whether to use DSCC or the command line to perform a specific task, see "Deciding When to Use DSCC and When to Use the Command Line" on page 33.

For more information about the administration framework, see "Directory Server Enterprise Edition Administration Model" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

This chapter covers the following topics:

Using DSCC for Directory Proxy Server

This section describes how to access DSCC for Directory Proxy Server.

▼ To Access DSCC for Directory Proxy Server

1 **Access DSCC in the same way as you would for Directory Server.**

See "To Access DSCC" on page 538.

2 **Click on the Proxy Server tab to view and manage Directory Proxy Server.**

The following figure shows the initial window for Directory Proxy Server.

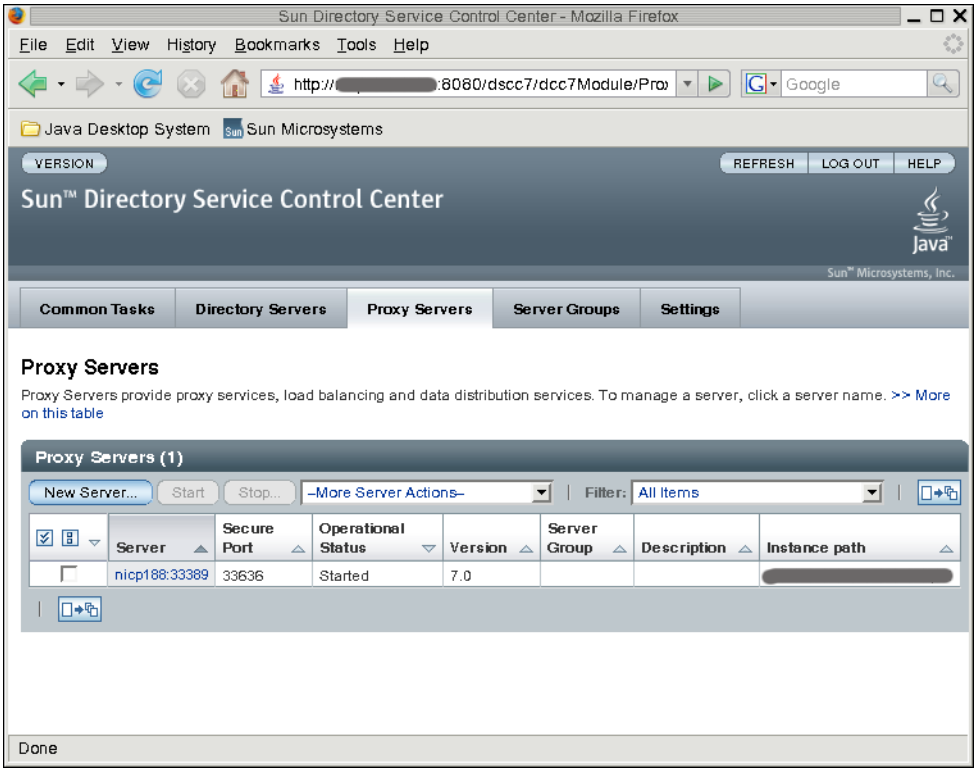


FIGURE 16–1 Initial DSCC Window for Directory Proxy Server

3 **Click a Directory Proxy Server instance to view or to manage that server.**

Note – For more information about using DSCC, see the online help.

Command-Line Tools for Directory Proxy Server

The commands-line tools that you use to work with Directory Proxy Server are called `dpadm` and `dpconf`. For information about how to use these commands, see the `dpadm`(1M) and `dpconf`(1M) man pages.

The `dpconf` is an LDAP based command so you must specify the user bind DN and password for the command to authenticate. While the `dpadm` command operates on the instance files.

This section describes the location of the `dpadm` and `dpconf` commands. It also provides information on environment variables, comparisons between the commands, and where to find help for using the commands.

Location of Directory Proxy Server Commands

The Directory Proxy Server command-line tools are located in the following directory by default:

install-path`/bin`

Your installation path depends on your operating system. Installation paths for all operating systems are listed in "Default Paths and Command Locations" on page 25.

Setting Environment Variables for `dpconf`

The `dpconf` command requires some options that you can preset by using environment variables. If you do not specify an option when using the command, or do not set the environment variable, the default setting will be used. You can configure environment variables for the following options:

`-D` *userDN*	User bind DN. Environment variable: `LDAP_ADMIN_USER`. Default: `cn=Proxy Manager`.
`-w` *password-file*	Password file for the user bind DN. Environment variable: `LDAP_ADMIN_PWF`. Default: Prompt for password.
`-h` *host*	Host name or IP address. Environment variable: `DIR_PROXY_HOST`. Default: `localhost`.
`-p` *LDAP-port*	LDAP port number. Environment variable: `DIR_PROXY_PORT`. Default: `389` if the server instance is running as `root`, and `1389` if the server instance is running as a regular user.
`-e`, `--unsecured`	Specifies that `dpconf` should open a clear connection by default. Environment variable: `DIR_PROXY_UNSECURED`. If this variable is not set, `dpconf` opens a secure connection by default.

For more details, see the `dpconf(1M)` man page.

Comparison of `dpadm` and `dpconf`

The following table shows a comparison of the `dpadm` and `dpconf` commands.

TABLE 16–1 Comparison of the `dpadm` and `dpconf` Commands

	`dpadm` **Command**	`dpconf` **Command**
Purpose	To manage the process or the files on a local instance of Directory Proxy Server	To configure a local or remote instance of Directory Proxy Server
User	Operating system user	LDAP user
Local or remote	The command *must* be local to the instance, that is, the command must be run on the host on which the server is running.	The command *can* be local to the instance but can also be run from anywhere on the network.
Example uses of the command	Create an instance of Directory Proxy Server. Start and stop an instance of Directory Proxy Server. Manage the certificate database.	Modify the configuration of an instance of Directory Proxy Server. Create a data view. Configure load balancing in a data source pool.
Server state	The server can be running or stopped.	The server *must* be running.
How the command identifies the server instance	By specifying the instance path. The instance path can be relative or absolute.	By specifying the host name or IP address and the port number. The command uses the LDAP port (`-p`) or the LDAPS secure port (`-P`). If a port number is not specified on the command line, the environment variable PROXY_PORT is used. If the environment variable is not set, the default ports are used.

Setting Multi-Valued Properties With `dpconf`

Certain Directory Proxy Server properties can take multiple values. Use the following syntax to specify the following values:

```
$ dpconf set-container-prop -h host -p port \
 property:value [property:value]
```

For example, to set multiple writable attributes for an LDAP data view named `my-view`, type the following command:

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 view-name\
 writable-attr:uid writable-attr:cn writable-attr:userPassword
```

To add a value to a multi-valued property that already contains values, type the following command:

```
$ dpconf set-container-prop -h host -p port \
 property+:value
```

To remove a value from a multi-valued property that already contains values, type the following command:

```
$ dpconf set-container-prop -h host -p port\
 property-:value
```

For example, in the scenario described previously, to add `sn` to the list of writable attributes, type the following command:

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 view-name\
writable-attr+:sn
```

To remove `cn` from the list of writable attributes, type the following command:

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 view-name\
writable-attr-:cn
```

Obtaining Help for Using `dpadm` and `dpconf`

For information about how to use the `dpadm` and `dpconf` commands, see the `dpadm(1M)` and `dpconf(1M)` man pages.

- To obtain a list of subcommands, type the appropriate command:

  ```
  $ dpadm --help
  ```

  ```
  $ dpconf --help
  ```

- To obtain information about how to use a subcommand, type the appropriate command:

  ```
  $ dpadm subcommand --help
  ```

  ```
  $ dpconf subcommand --help
  ```

- To obtain information about the configuration properties used in the `dpconf` command, type:

  ```
  $ dpconf help-properties
  ```

- To obtain information about the configuration properties for a subcommand, use this command:

  ```
  $ dpconf help-properties subcommand-entity
  ```

  For example, to find information about the access log properties, type:

```
$ dpconf help-properties access-log
```

- To obtain information about a property used in a subcommand, use this command:

  ```
  $ dpconf help-properties subcommand-entity property
  ```

  For example, to find information about the `log-search-filters` property of the `set-access-log-prop` subcommand, type:

  ```
  $ dpconf help-properties access-log log-search-filters
  ```

- To list the key properties of a group of entities, such as data views or connection handlers, use the verbose option `-v` with the `list` subcommand.

  For example, to view the key properties and relative priorities of all of the connection handlers, use this command:

  ```
  $ dpconf list-connection-handlers -h host -p port -v
  Name                        is-enabled  priority  description
  --------------------------  ----------  --------  --------------------------
  anonymous                   false       99        unauthenticated connections
  default connection handler  true        100       default connection handler
  dscc administrators        true          1         Administrators connection
                                                     handler
  ```

  For more information about an individual property, see the man page corresponding to that property.

◆ ◆ ◆ CHAPTER 17

Directory Proxy Server Instances

This chapter describes how to administer an instance of Directory Proxy Server. This chapter covers the following topics:

Working With Directory Proxy Server Instances

When you create an instance of Directory Proxy Server, the files and directories required for the instance are created in the path that you specify.

In this procedure, you create a *server instance* on the local host using the `dpadm` command. You then configure the instance using the `dpconf` command.

Non-`root` users can create server instances.

A Directory Proxy Server instance must be configured to proxy directory client application requests to *data sources* through *data views*. When you start or stop an instance, you start or stop the server process that proxies directory client application requests.

The `dpadm` command enables you to manage a Directory Proxy Server instance and the files belonging to that instance on the local host. The command does not allow you to administer servers over the network, but only directly on the local host. The `dpadm` command has subcommands for each key management task. For a complete description, see `dpadm(1M)`.

The `dpconf` command is an LDAP client. The command enables you to configure nearly all server settings on a running Directory Proxy Server instance from the command line. You can configure settings whether the server is on the local host or another host that is accessible across the network. The `dpconf` command has subcommands for each key configuration task. For a complete description, see `dpconf(1M)`.

▼ To Create a Directory Proxy Server Instance

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

If you use DSCC to create a new server instance, you can choose to copy some or all of the server configuration settings from an existing server.

1 Create the instance of Directory Proxy Server.

```
$ dpadm create -p port instance-path
```

For example, to create a new instance in the directory `/local/dps`, use this command:

```
$ dpadm create -p 2389 /local/dps
```

To specify any other parameter of the instance, see the `dpadm(1M)` man page.

2 Type a password if required.

3 Confirm that the instance has been created by verifying the status of the instance.

```
$ dpadm info instance-path
```

4 (Optional) If you installed Directory Proxy Server using the native packages, and your operating provides a service management solution, you can enable the server to be managed as a service, as shown in this table.

Operating System	Command
Solaris 10	`dpadm enable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart` *instance-path*
Windows	`dpadm enable-service --type WIN_SERVICE` *instance-path*

5 (Optional) Register the server instance with Directory Service Control Center by using either of the following methods.

- **Login to DSCC, and then use the Register Existing Server action on the Proxy Servers tab.**

  Access DSCC using `http://`*hostname*`:8080/dscc7` or `https://`*hostname*`:8181/dscc7` as per your application server configuration.

- **Use the command `dsccreg add-server`.**

```
$ dsccreg add-server -h hostname --description "My Proxy" /local/dps
Enter DSCC administrator's password:
/local/dps is an instance of DPS
Enter password of "cn=Proxy Manager" for /local/dps:
```

```
Connecting to /local/dps
Enabling DSCC access to /local/dps
Registering /local/dps in DSCC on hostname.
```

See `dsccreg(1M)` for more information about the command.

To Find the Status of a Directory Proxy Server Instance

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Find the status of an instance of Directory Proxy Server.**

  `$ dpadm info` *instance-path*

To Start and Stop Directory Proxy Server

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **To start or stop Directory Proxy Server, do one of the following.**
  - To start Directory Proxy Server, type:

    `$ dpadm start` *instance-path*

    For example, to start an instance at `/local/dps`, use this command:

    ```
    $ dpadm start /local/dps
    ```

  - To stop Directory Proxy Server, type:

    `$ dpadm stop` *instance-path*

    For example:

    ```
    $ dpadm stop /local/dps
    ```

To List All the Running Instances

- **List the running instances on a host using the following command:**

  ```
  dpadm list-running-instances [--all]
  ```

  The `--all` option lists the running instances from any installation path.

To Stop the Running Instances

- **Stop the running instances on a host using the following command:**

  ```
  dpadm stop-running-instances [-i] [--force]
  ```

  For more information, see dpadm(1M).

To View Whether It Is Necessary to Restart a Directory Proxy Server Instance

Sometimes, a configuration change requires the server to be restarted before the change takes effect. Use this procedure to check whether it is necessary to restart a Directory Proxy Server instance after a configuration change.

- **View whether it is necessary to restart the server.**

  ```
  $ dpconf get-server-prop -h host -p port is-restart-required
  ```

  - If the command returns `true`, you must restart the instance of Directory Proxy Server.
  - If the command returns `false`, it is not necessary to restart the instance of Directory Proxy Server.

To Restart Directory Proxy Server

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Restart Directory Proxy Server.**

  ```
  $ dpadm restart instance-path
  ```

  For example, to restart an instance at `/local/dps`, use this command:

  ```
  $ dpadm restart /local/dps
  ```

To Delete a Directory Proxy Server Instance

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **(Optional) Stop the Directory Proxy Server instance.**

   ```
   $ dpadm stop instance-path
   ```

If you do not stop the instance, the delete command will stop it automatically. However, if you have enabled the instance in a service management solution, you must stop it manually.

2 **(Optional) If you have previously used DSCC to manage the server, use the command line to unregister the server.**

```
$ dsccreg remove-server /local/dps
Enter DSCC administrator's password:
/local/dps is an instance of DPS
Enter password of "cn=Proxy Manager" for /local/dps:
Unregistering /local/dps from DSCC on localhost.
Connecting to /local/dps
Disabling DSCC access to /local/dps
```

For details, see the `dsccreg(1M)` man page.

3 **(Optional) If you previously enabled the server instance in a service management solution, then disable the server from being managed as a service.**

Operating System	Command
Solaris 10	`dpadm disable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart --off` *instance-path*
Windows	`dpadm disable-service --type WIN_SERVICE` *instance-path*

4 **Delete the instance.**

```
$ dpadm delete instance-path
```

Configuring Directory Proxy Server Instances

This section describes how to configure an instance of Directory Proxy Server. The procedures in this section use the `dpadm` and `dpconf` commands. For information about these commands, see the `dpadm(1M)` and `dpconf(1M)` man pages.

▼ To Display the Configuration of Directory Proxy Server Instance

- **Type** `dpconf info`

```
$ dpconf info -p port
Instance Path           :   instance path
Host Name               :   host
```

```
Secure listen address   :  IP address
Port                    :  port
Secure port             :  secure port
SSL server certificate  :  defaultServerCert

Directory Proxy Server needs to be restarted.
```

`dpconf info` displays `Secure listen address` and `Non-secure listen address` only if these properties are set to non-default values. The above output does not display `Non-secure listen address`, as this property is not set to a non-default value.

`dpconf info` also reminds the user to restart the instance if it needs to be restarted.

You can also use `dpadm info` *INSTANCE_PATH* to display Directory Proxy Server instance configuration information.

▼ To Modify the Configuration of Directory Proxy Server

This section describes how to modify the configuration of Directory Proxy Server.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Find the current configuration of Directory Proxy Server.

```
$ dpconf get-server-prop -h host -p port

allow-cert-based-auth                     : allow
allow-ldapv2-clients                      : true
allow-persistent-searches                 : false
allow-sasl-external-authentication        : true
allow-unauthenticated-operations          : true
allow-unauthenticated-operations-mode     : anonymous-and-dn-identified
allowed-ldap-controls                     : -
cert-data-view-routing-custom-list        : none
cert-data-view-routing-policy             : all-routable
cert-search-attr-mappings                 : none
cert-search-base-dn                       : none
cert-search-bind-dn                       : none
cert-search-bind-pwd                      : none
cert-search-user-attr                     : userCertificate
compat-flag                               : none
configuration-manager-bind-dn             : cn=proxy manager
configuration-manager-bind-pwd            : {3DES}RPdIFbvoWdvhLR8lU43zCMZyKFGPxfFg
connection-pool-wait-timeout              : 3s
data-source-read-timeout                  : 20s
data-view-automatic-routing-mode          : automatic
email-alerts-enabled                      : false
```

```
email-alerts-message-from-address          : local
email-alerts-message-subject               : Proxy Server Administrative Alert
email-alerts-message-subject-includes      : false
-alert-code
email-alerts-message-to-address            : root@localhost
email-alerts-smtp-host                     : localhost
email-alerts-smtp-port                     : smtp
enable-remote-user-mapping                 : false
enable-user-mapping                        : false
enabled-admin-alerts                       : all
enabled-ssl-cipher-suites                  : JRE
enabled-ssl-protocols                      : SSLv3
enabled-ssl-protocols                      : TLSv1
encrypt-configuration                      : true
extension-jar-file-url                     : none
is-restart-required                        : false
number-of-search-threads                   : 20
number-of-worker-threads                   : 50
proxied-auth-check-timeout                 : 30m
remote-user-mapping-bind-dn-attr           : none
revert-add-on-failure                      : true
scriptable-alerts-command                  : echo
scriptable-alerts-enabled                  : false
search-mode                                : sequential
search-wait-timeout                        : 10s
ssl-client-cert-alias                      : none
ssl-server-cert-alias                      : defaultServerCert
supported-ssl-cipher-suites                : SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                : SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                : SSL_DHE_DSS_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                : SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                : SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                : SSL_DHE_RSA_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                : SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                : SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
supported-ssl-cipher-suites                : SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                : SSL_DH_anon_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                : SSL_DH_anon_WITH_RC4_128_MD5
supported-ssl-cipher-suites                : SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                : SSL_RSA_EXPORT_WITH_RC4_40_MD5
supported-ssl-cipher-suites                : SSL_RSA_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                : SSL_RSA_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                : SSL_RSA_WITH_NULL_MD5
supported-ssl-cipher-suites                : SSL_RSA_WITH_NULL_SHA
supported-ssl-cipher-suites                : SSL_RSA_WITH_RC4_128_MD5
supported-ssl-cipher-suites                : SSL_RSA_WITH_RC4_128_SHA
supported-ssl-cipher-suites                : TLS_DHE_DSS_WITH_AES_128_CBC_SHA
supported-ssl-cipher-suites                : TLS_DHE_RSA_WITH_AES_128_CBC_SHA
```

```
supported-ssl-cipher-suites                : TLS_DH_anon_WITH_AES_128_CBC_SHA
supported-ssl-cipher-suites                : TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5
supported-ssl-cipher-suites                : TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA
supported-ssl-cipher-suites                : TLS_KRB5_EXPORT_WITH_RC4_40_MD5
supported-ssl-cipher-suites                : TLS_KRB5_EXPORT_WITH_RC4_40_SHA
supported-ssl-cipher-suites                : TLS_KRB5_WITH_3DES_EDE_CBC_MD5
supported-ssl-cipher-suites                : TLS_KRB5_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                : TLS_KRB5_WITH_DES_CBC_MD5
supported-ssl-cipher-suites                : TLS_KRB5_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                : TLS_KRB5_WITH_RC4_128_MD5
supported-ssl-cipher-suites                : TLS_KRB5_WITH_RC4_128_SHA
supported-ssl-cipher-suites                : TLS_RSA_WITH_AES_128_CBC_SHA
supported-ssl-protocols                    : SSLv2Hello
supported-ssl-protocols                    : SSLv3
supported-ssl-protocols                    : TLSv1
syslog-alerts-enabled                      : false
syslog-alerts-facility                     : USER
syslog-alerts-host                         : localhost
time-resolution                            : 250ms
time-resolution-mode                       : custome-resolution
use-cert-subject-as-bind-dn                : true
use-external-schema                        : false
user-mapping-anonymous-bind-dn             : none
user-mapping-anonymous-bind-pwd            : none
user-mapping-default-bind-dn               : none
user-mapping-default-bind-pwd              : none
verify-certs                               : false
```

Alternatively, view the current setting of one or more configuration properties.

```
$ dpconf get-server-prop -h host -p port property-name ...
```

For example, find whether unauthenticated operations are allowed by running this command:

```
$ dpconf get-server-prop -h host -p port allow-unauthenticated-operations
allow-unauthenticated-operations  :  true
```

2 Change one or more of the configuration parameters.

```
$ dpconf set-server-prop -h host -p port property:value ...
```

For example, disallow unauthenticated operations by running this command:

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
```

If you attempt to perform an illegal change, the change is not made. For example, if you set the `allow-unauthenticated-operations` parameter to `f` instead of `false`, the following error is produced:

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:f
The value "f" is not a valid value for the property "allow-unauthenticated-operations".
Allowed property values: BOOLEAN
The "set-server-prop" operation failed.
```

3 If necessary, restart the instance of Directory Proxy Server for the changes to take effect.

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Configuring the Proxy Manager

The Proxy Manager is the privileged administrator, comparable to the root user on UNIX® systems. The Proxy Manager entry is defined when an instance of Directory Proxy Server is created. The default DN of the Proxy Manager is `cn=Proxy Manager`.

You can view and change the Proxy Manager DN and password, as shown in the following procedure.

▼ To Configure the Proxy Manager

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Find the configuration of the Proxy Manager.

```
$ dpconf get-server-prop -h host -p port configuration-manager-bind-dn\
 configuration-manager-bind-pwd


configuration-manager-bind-dn   :  cn=proxy manager
configuration-manager-bind-pwd  :  {3DES}U77v39WX8MDpcWVrueetB0lfJlBc6/5n
```

The default value for the Proxy Manager is `cn=proxy manager`. A hashed value is returned for the configuration manager password.

2 Change the DN of the Proxy Manager.

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-dn:bindDN
```

3 Create a file that contains the password for the Proxy Manager and set the property that points to that file.

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-pwd-file:filename
```

Configuration Changes Requiring Server Restart

Most configuration changes to Directory Proxy Server and its entities can be made online. Certain changes require that the server be restarted before the changes take effect. If you make configuration changes to any properties in the following list, the server must be restarted:

```
custom-distribution-algorithm
distribution-algorithm
db-name
db-url
db-user
custom-distribution-algorithm
distribution-algorithm
custom-distribution-algorithm
distribution-algorithm
bind-dn
client-cred-mode
ldap-address
ldap-port
ldaps-port
num-bind-init
num-read-init
num-write-init
ssl-policy
load-balancing-algorithm
custom-distribution-algorithm
distribution-algorithm
listen-address
listen-port
number-of-threads
listen-address
listen-port
number-of-threads
custom-distribution-algorithm
distribution-algorithm
compat-flag
number-of-search-threads
number-of-worker-threads
syslog-alerts-enabled
syslog-alerts-host
time-resolution
use-external-schema
aci-data-view
```

The rws and rwd keywords of a property indicate whether changes to the property require the server to be restarted.

- If a property has an rws (read, write, static) keyword, the server must be restarted when the property is changed.
- If a property has an rwd (read, write, dynamic) keyword, modifications to the property are implemented dynamically (without restarting the server).

To determine whether a change to a property requires the server to be restarted, run the following command:

```
$ dpconf help-properties | grep property-name
```

For example, to determine whether changing the bind DN of an LDAP data source requires the server to be restarted, run the following command:

```
$ dpconf help-properties | grep bind-dn
connection-handler       bind-dn-filters        rwd  STRING | any
This property specifies a set of regular expressions. The bind DN
of a client must match at least one regular expression in order for
the connection to be accepted by the connection handler. (Default: any)
ldap-data-source      bind-dn               rws  DN | ""
This property specifies the DN to use when binding to the LDAP data
source. (Default: undefined)
```

To determine whether the server must be restarted following a configuration change, run the following command:

```
$ dpconf get-server-prop -h host -p port is-restart-required
```

Backing Up and Restoring Directory Proxy Server Instances

When you use dpadm to back up Directory Proxy Server, the configuration files and server certificates are backed up. If you have implemented Directory Proxy Server virtual ACIs, the ACIs are also backed up.

Directory Proxy Server automatically backs up the conf.ldif file whenever the server starts successfully.

▼ To Back Up a Directory Proxy Server Instance

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Stop the instance of Directory Proxy Server.**

   ```
   $ dpadm stop instance-path
   ```

2. **Back up the instance of Directory Proxy Server.**

   ```
   $ dpadm backup instance-path archive-dir
   ```

   The *archive-dir* directory is created by the `backup` command and must not exist before you run the command. This directory contains a backup of each of the configuration files and the certificates.

▼ To Restore a Directory Proxy Server Instance

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

You must create a Directory Proxy Server instance before starting the restore operation.

1. **Stop the instance of Directory Proxy Server.**

   ```
   $ dpadm stop instance-path
   ```

2. **Restore the instance of Directory Proxy Server.**

   ```
   $ dpadm restore instance-path archive-dir
   ```

   - If the instance path exists, the restore operation is performed silently. The configuration files and the certificates in the *archive-dir* directory replace those in the *instance-path* directory.
   - If the instance path does not exist, the restore operation fails.

◆ ◆ ◆ CHAPTER 18

LDAP Data Views

An LDAP data view exposes data in an LDAP server to a client request and specifies the data source pool that responds to the request. By defining LDAP data views, you can perform the following tasks:

- Expose a whole database in a single view
- Provide different views for different subtrees in a database
- Provide a unified view of different databases

Creating LDAP Data Views

Creating an LDAP data view includes the following steps:

1. "To Create an LDAP Data Source" on page 373.
2. "To Create an LDAP Data Source Pool" on page 376.
3. "To Attach an LDAP Data Source to a Data Source Pool" on page 378.
4. "To Create an LDAP Data View" on page 379.

Creating and Configuring LDAP Data Sources

This section describes how to use the `dpconf` command to create and configure LDAP data sources. For reference information about these topics, see "LDAP Data Sources" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

For information about how to create and configure LDAP data sources, see the following procedures.

▼ To Create an LDAP Data Source

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Create the data source.**

   ```
   $ dpconf create-ldap-data-source -h host -p port [-s] source-name host:port
   ```

   In this command, *source-name* is a name that you assign to the new data source. *host* and *port* refer to the host and port on which the LDAP server is running. Note that the data source does not use SSL by default. Use `-s` to enable to specify a secure port.

   If the host is specified by an IP V6 address, you need to use the IP V6 reference when you create the data source. For example, if Directory Proxy Server will bind to a host with the IP V6 address `fe80::209:3dff:fe00:8c93` on port `2389`, use the following command to create the data source:

   ```
   $ dpconf create-ldap-data-source -h host1 -p 1389 ipv6-host \
    [fe80::209:3dff:fe00:8c93]:2389
   ```

   If you use the console to create the data source, you must specify the actual IP V6 address (without the square brackets).

   For information about how to modify the properties of an LDAP data source, see "To Configure an LDAP Data Source" on page 374.

2. **(Optional) View the list of data sources.**

   ```
   $ dpconf list-ldap-data-sources -h host -p port
   ```

▼ To Configure an LDAP Data Source

The following procedure shows how to display the properties of an LDAP data source and how to set the properties that you require to change. The procedure shows the commands using which any of the properties of the LDAP data source can be changed. It also shows how to get the detailed information of a property, which helps you to set that property.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **View the properties of the data source by using this command syntax:**

   ```
   $ dpconf get-ldap-data-source-prop -h host -p port \
   [-M unit] [-Z unit] source-name [property...]
   ```

   In this command, `-M` and `-Z` refer to the units in which you want data to be displayed. The `M` option specifies the unit of time. The value for `-M` can be `M`, `w`, `d`, `h`, `m`, `s`, or `ms`, to represent months, weeks, days, hours, minutes, seconds, or milliseconds. The `-Z` option specifies the data size unit. The value for `-Z` can be `T`, `G`, `M`, `k`, or `b`, to represent Terabytes, Gigabytes, Megabytes, kilobytes, or bytes.

If you do not specify a property, all properties are displayed. The default properties of an LDAP data source are as follows:

```
bind-dn                          :  -
bind-pwd                         :  -
client-cred-mode                 :  use-client-identity
connect-timeout                  :  10s
description                      :  -
down-monitoring-interval         :  inherited
is-enabled                       :  false
is-read-only                     :  true
ldap-address                     :  host
ldap-port                        :  port
ldaps-port                       :  ldaps
monitoring-bind-timeout          :  5s
monitoring-entry-dn              :  ""
monitoring-entry-timeout         :  5s
monitoring-inactivity-timeout    :  2m
monitoring-interval              :  30s
monitoring-mode                  :  reactive
monitoring-retry-count           :  3
monitoring-search-filter         :  (|(objectClass=*)(objectClass=ldapSubEntry))
num-bind-incr                    :  10
num-bind-init                    :  10
num-bind-limit                   :  1024
num-read-incr                    :  10
num-read-init                    :  10
num-read-limit                   :  1024
num-write-incr                   :  10
num-write-init                   :  10
num-write-limit                  :  1024
proxied-auth-check-timeout       :  1.8s
proxied-auth-use-v1              :  false
ssl-policy                       :  never
use-read-connections-for-writes  :  false
use-tcp-keep-alive               :  true
use-tcp-no-delay                 :  true
```

2 Enable the data source.

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-enabled:true
```

3 Configure all the properties that are listed in Step 1, if you want to change the default settings.

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name property:value
```

For example, if you want to modify entries on a data source, configure the data source to allow write operations.

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-read-only:false
```

To use a read connection to process write operation when all the write connections are busy, run the following command. The vice-versa is also true.

```
dpconf set-ldap-data-source-prop -h host -p port source-name \
use-read-connections-for-writes:true
```

To find information about a property used in a subcommand, run this command:

```
$ dpconf help-properties ldap-data-source property
```

For example, to find information about the `is-read-only` property, run this command:

```
dpconf help-properties ldap-data-source is-read-only
```

To list the key properties for data sources, use the verbose option `-v` with the `list-ldap-data-sources` subcommand.

```
$ dpconf list-ldap-data-sources -v
Name         is-enabled  ldap-address  ldap-port  ldaps-port  description
-----------  ----------  ------------  ---------  ----------  -----------
datasource0  true        myHost        myPort     ldaps       -
datasource1  true        myHost        myPort     ldaps       -
```

4 If necessary, restart the instance of Directory Proxy Server for the changes to take effect.

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364. For a list of configuration changes that require a server restart, see "Configuration Changes Requiring Server Restart" on page 370.

Creating and Configuring LDAP Data Source Pools

This section describes how to use the `dpconf` command to create and configure LDAP data source pools. For reference information about these topics, see "LDAP Data Sources" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

For information about how to create and configure data source pools, see the following procedures:

▼ To Create an LDAP Data Source Pool

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Create one or more data source pools.

```
$ dpconf create-ldap-data-source-pool -h host -p port pool-name
```

Additional data source pools can be specified after the first *pool-name*. For information about how to modify the properties of a data source pool, see "To Configure an LDAP Data Source Pool" on page 377.

2 (Optional) View the list of data source pools.

```
$ dpconf list-ldap-data-source-pools -h host -p port
```

▼ To Configure an LDAP Data Source Pool

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the properties of the data source pool by using this command syntax:

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port \
[-M unit] [-Z unit] pool-name [property...]
```

In this command, `-M` and `-Z` refer to the units in which you want data to be displayed. The `M` option specifies the unit of time. The value for `-M` can be `M`, `w`, `d`, `h`, `m`, `s`, or `ms`, to represent months, weeks, days, hours, minutes, seconds, or milliseconds. The `-Z` option specifies the data size unit. The value for `-Z` can be `T`, `G`, `M`, `k`, or `b`, to represent Terabytes, Gigabytes, Megabytes, kilobytes, or bytes.

If you do not specify a property, all properties are displayed. The default properties of an LDAP data source pool are as follows:

```
client-affinity-bind-dn-filters     :  any
client-affinity-criteria            :  connection
client-affinity-ip-address-filters  :  any
client-affinity-policy              :  write-affinity-after-write
client-affinity-timeout             :  20s
description                         :  -
enable-client-affinity              :  false
load-balancing-algorithm            :  proportional
```

2 Configure the properties that are listed in Step 1.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 property:value
```

For information about how to configure the properties of a data source pool for load balancing and client affinity, see Chapter 20, "Directory Proxy Server Load Balancing and Client Affinity."

Attaching LDAP Data Sources to a Data Source Pool

A data source that is attached to a data source pool is called an *attached data source*. The properties of an attached data source determine the load balancing configuration of the data source pool. When you configure the weights of an attached data source, consider the weights of

all of the attached data sources in a data source pool. Ensure that the weights work together as required. For information about how to configure weights for load balancing, see "To Configure Weights for Load Balancing" on page 406.

▼ To Attach an LDAP Data Source to a Data Source Pool

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Attach one or more data sources to a data source pool.

```
$ dpconf attach-ldap-data-source -h host -p port pool-name \
 source-name [source-name ...]
```

2 (Optional) View the list of attached data sources for a given data source pool.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -E pool-name
```

In this command, `-E` is optional, and modifies the display output to show one property value per line.

3 (Optional) View the key properties of the attached data sources for a given data source pool.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

In this command, `-v` specifies verbose output. For example, view the properties of an example data source pool.

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v My-pool
SRC_NAME     add-weight  bind-weight  compare-weight
-----------  ----------  -----------  --------------
datasource0  disabled    disabled     disabled
datasource1  disabled    disabled     disabled

delete-weight  modify-dn-weight  modify-weight  search-weight
-------------  ----------------  -------------  -------------
disabled       disabled          disabled       disabled
disabled       disabled          disabled       disabled
```

4 (Optional) View the properties of an attached data source by using the following command syntax:

```
$ dpconf get-attached-ldap-data-source-prop -h host -p port [-M unit] [-Z unit] \
 pool-name source-name [property...]
```

In this command, `-M` and `-Z` refer to the units in which you want data to be displayed. The `M` option specifies the unit of time. The value for `-M` can be `M`, `w`, `d`, `h`, `m`, `s`, or `ms`, to represent months, weeks, days, hours, minutes, seconds, or milliseconds. The `-Z` option specifies the data size unit. The value for `-Z` can be `T`, `G`, `M`, `k`, or `b`, to represent Terabytes, Gigabytes, Megabytes, kilobytes, or bytes.

If you do not specify a property, all properties are displayed.

The properties of an attached data source define the weight for each type of operation in load balancing. The default weights of an attached data source are as follows:

```
add-weight        :  disabled
bind-weight       :  disabled
compare-weight    :  disabled
delete-weight     :  disabled
modify-dn-weight  :  disabled
modify-weight     :  disabled
search-weight     :  disabled
```

5 The default weights of an attached data source are 0 and disabled. You must set the weights of an attached data source for Directory Proxy Server to work as intended.

In the following example, all the properties are set to one. You can change the values of these properties as per your requirements. For information about how to configure weights of an attached data source for load balancing, see "To Configure Weights for Load Balancing" on page 406.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name add-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name bind-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name compare-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name delete-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name modify-dn-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name modify-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name search-weight:1
```

Working with LDAP Data Views

For information about how to create and configure LDAP data views, see the following procedures:

▼ To Create an LDAP Data View

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Create an LDAP data view.

```
$ dpconf create-ldap-data-view -h host -p port view-name pool-name suffix-DN
```

For information about how to modify the properties of an LDAP data view, see "To Configure an LDAP Data View" on page 380.

2 View the list of LDAP data views.

```
$ dpconf list-ldap-data-views -h host -p port
```

▼ To Configure an LDAP Data View

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the properties of an LDAP data view.

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name
```

If you create a data view without configuring any of the properties, your data view has the following configuration:

```
alternate-search-base-dn                    :  ""
attr-name-mappings                          :  none
base-dn                                     :  suffix-DN
contains-shared-entries                     :  false
custom-distribution-algorithm-class         :  none
description                                 :  -
distribution-algorithm                      :  none
dn-join-rule                                :  none
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  none
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  none
non-writable-attr                           :  none
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-base-dn-regular-expression :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
```

```
process-bind                              :  -
replication-role                          :  master
viewable-attr                             :  all except non-viewable-attr
writable-attr                             :  all except non-writable-attr
```

Note – All users except the Proxy Manager see the `cn=config` and `cn=monitor` suffixes from the back-end server. By default, data from the back-end servers is not available to the Proxy Manager. The `cn=config` and `cn=monitor` subtrees that are available to the Proxy Manager are those of the proxy itself.

When you create a Directory Proxy Server instance, a connection handler for the Proxy Manager is created with an empty data view policy. If the Proxy Manager requires access to back-end data, you must add a data view to the data view policy of the Proxy Manager connection handler. On such a data view, the `cn=config` and `cn=monitor` subtrees are excluded by default.

2 Change one or more of the properties that are listed in Step 1.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 property:value [property:value ... ]
```

For example, to access the `dc=example,dc=com` subtree on a data source, specify `dn-mapping-source-base-dn` in the data view.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
dn-mapping-source-base-dn:dc=example,dc=com
```

To add a value to a multi-valued property, use this command:

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name property+:value
```

To remove a value from a multi-valued property, use this command:

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name property-:value
```

3 If necessary, restart the instance of Directory Proxy Server for the changes to take effect.

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Accessing Configuration Entries for a Directory Server by Using Directory Proxy Server

The configuration entries for Directory Proxy Server are in `cn=config`. When you use Directory Proxy Server to access configuration entries, by default, you access the configuration entries of Directory Proxy Server.

To access the configuration entries of a directory server, it is better to connect directly to Directory Server, not to Directory Proxy Server. For information about how to configure Directory Server, see Chapter 3, "Directory Server Configuration."

Caution – If you reconfigure Directory Proxy Server to access the configuration entries of a directory server, you are likely to break the administration framework of Directory Proxy Server.

If you really need to access the configuration entries of a directory server through Directory Proxy Server, take special steps to ensure that you do not break the administration framework of Directory Proxy Server. This section describes how to access the configuration entries of a directory server by using Directory Proxy Server.

▼ To Access the Configuration Entries of a Directory Server by Using Directory Proxy Server

1. **Create a data source as described in "Creating and Configuring LDAP Data Sources" on page 373.**

2. **Create an LDAP data source pool as described in "Creating and Configuring LDAP Data Source Pools" on page 376.**

3. **To expose the configuration entries of one specific data source, attach only one LDAP data source to the LDAP data source pool as described in "Attaching LDAP Data Sources to a Data Source Pool" on page 377.**

   ```
   $ dpconf attach-ldap-data-source -h host -p port pool-name data-source-name
   ```

   After performing this step, a client can access the configuration entries of the data source that is connected to Directory Proxy Server.

   If you attach more than one LDAP data source to the LDAP data source pool, you can access the configuration entries of one of the data sources connected to Directory Proxy Server. However, you cannot know which data source the configuration entries belong to.

You must set the weights of an attached data source for Directory Proxy Server to work as intended. For more information, see "Attaching LDAP Data Sources to a Data Source Pool" on page 377.

4 **Create an LDAP data view to expose** `cn=config`**.**

```
$ dpconf create-ldap-data-view -h host -p port view-name pool-name cn=config
```

Renaming Attributes and DNs

Each entry in a directory is identified by a DN and a set of attributes and their values. Often, the DNs and the attributes defined on the client side do not map to the DNs and the attributes defined on the server side. Data views can be defined to rename DNs and attributes. When a client makes a request, the DNs and attributes are renamed to match the server side. When the result is returned to a client, the DNs and attributes are changed back to match the client side.

For information about attribute renaming and DN renaming, see "Attribute Renaming and DN Renaming" in *Sun Directory Server Enterprise Edition 7.0 Reference*. For information about how to rename attributes and DNs, see the following procedures:

▼ To Configure Attribute Renaming

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Set one or more** `attr-name-mappings` **properties on the data view for which you want to configure attribute mapping.**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  attr-name-mappings:client-side-attribute-name#server-side-attribute-name[#qualifier]\
  [attr-name-mappings:client-side-attribute-name#server-side-attribute-name#qualifier...]
```

For example, rename `surname` on the client side to `sn` on the server side.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 attr-name-mappings:surname#sn
```

To add an attribute mapping to an existing list of mappings, use this command:

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings+:client-side-attribute-name#server-side-attribute-name[#qualifier]
```

To remove an attribute mapping from an existing list of mappings, use this command:

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings-:client-side-attribute-name#server-side-attribute-name[#qualifier]
```

▼ To Configure DN Renaming

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the `base-dn` property and the DN mapping properties of the data view for which you want to rename DNs.

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

The properties have the following meanings:

- `base-dn` is the DN of the subtree on the client side, which is equivalent to the base DN of the data view.
- `dn-mapping-source-base-dn` is the DN of the subtree on the server side.
- `dn-mapping-attrs` defines a list of attributes that contain DNs of entries.

For example, the data view for the `dc=example,dc=com` database on the client side has the following values when DN renaming is not defined:

```
$ dpconf get-ldap-data-view-prop myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                  :  dc=example,dc=com
 dn-mapping-attrs         :  none
 dn-mapping-source-base-dn  :  none
```

2 Map a DN on the client side to a DN on the server side.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-source-base-dn:server-side-dn
```

For example, map the `dc=example,dc=com` database on the client side to `dc=example,dc=org` on the server side.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 dn-mapping-source-base-dn:dc=example,dc=org
```

3 Rename attributes in the portion of the DIT that is affected by Step 2, if those attributes contain DNs.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-attrs:attribute-name [dn-mapping-attrs:attribute-name ...]
```

For example, if the `group` attribute contains DNs in the namespace affected by the rename operation in Step 2, rename the attribute as follows:

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView dn-mapping-attrs:group
```

To add a DN mapping to an existing list of mappings, use this command:

```
$ dpconf set-ldap-data-view-prop -h host -p port \
view-name dn-mapping-attrs+:attribute-name
```

To remove a DN mapping from an existing list of mappings, use this command:

```
$ dpconf set-ldap-data-view-prop -h host -p port \
view-name dn-mapping-attrs-:attribute-name
```

4 **View the `base-dn` property and the DN mapping properties of the data view for which you have renamed DNs.**

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

For example, the data view for the `dc=example,dc=com` database on the client side has the following values after DN renaming:

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                   :  dc=example,dc=com
 dn-mapping-attrs          :  group
 dn-mapping-source-base-dn  :  dc=example,dc=org
```

Configuring View Exclusion Base and Alternate Search Base

When a subordinate data view is created, Directory Proxy Server automatically excludes the subordinate data view from the superior data view. When a request targets the subordinate data view, the request is sent to the subordinate data view, not to the superior data view.

When an alternate search base is specified in a subordinate data view, search operations targeted at the superior data view are also performed in the subordinate data view.

By default, Directory Proxy Server automatically configures the `excluded-subtrees` and `alternate-search-base-dn` properties. The following procedure describes how to configure these properties manually.

▼ To Manually Configure the `excluded-subtrees` and `alternate-search-base-dn` Properties

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Configure Directory Proxy Server to manually route requests.**

   ```
   $ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:manual
   ```

   When `data-view-automatic-routing-mode` is `manual`, Directory Proxy Server does not generate the `excluded-subtrees` and `alternate-search-base-dn` properties. You must set the values of these properties manually. The values that you set here are not checked by Directory Proxy Server. Be aware that setting these values incorrectly can break the administration path.

   Alternatively, configure Directory Proxy Server to partially route requests manually.

   ```
   $ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:limited
   ```

   When `data-view-automatic-routing-mode` is `limited`, Directory Proxy Server does not generate the `excluded-subtrees` and `alternate-search-base-dn` properties. However, Directory Proxy Server does check that the values set here do not conflict with the administration path.

2. **Configure the view exclusion base.**

   ```
   $ dpconf set-ldap-data-view-prop -h host -p port view-name excluded-subtrees:suffix-DN
   ```

   The view exclusion base determines branches of the DIT whose entries are not exposed by the data view.

3. **Configure the alternate search base.**

   ```
   $ dpconf set-ldap-data-view-prop -h host -p port view-name \
    alternate-search-base-dn:search-base-DN
   ```

   The alternate search base determines other branches of the DIT in which entries belonging to this data view may be located. The base DN is defined by default as an alternate search base in all data views.

Creating and Configuring Data Views for Example Use Cases

This section contains the following information about data views and how to create and configure them:

The examples in this section assume that the connection handler allows all client connections to be processed by Directory Proxy Server.

Default Data View

If you create a data view without configuring any of the properties, your data view has the following configuration:

```
alternate-search-base-dn                    :  ""
alternate-search-base-dn                    :  base-DN
attr-name-mappings                          :  none
base-dn                                     :  suffix-DN
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

Data Views That Route All Requests, Irrespective of the Target DN of the Request

This section shows the configuration of a data view that routes all the requests that do not match their targets mentioned in their target DNs, to a data source pool. This data view is called the *root data view*. The root data view is created by default when an instance of Directory Proxy Server is created. For information about the root data view, see "Data Views to Route All Requests, Irrespective of the Target DN of the Request" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The root data view has the following configuration:

```
alternate-search-base-dn                  :  -
attr-name-mappings                        :  none
base-dn                                   :  ""
contains-shared-entries                   :  -
description                               :  Automatically-generated data view
                                             able to route client operations
                                             independently of the operation base dn
distribution-algorithm                    :  -
dn-join-rule                              :  -
dn-mapping-attrs                          :  none
dn-mapping-source-base-dn                 :  none
excluded-subtrees                         :  ""
excluded-subtrees                         :  cn=config
excluded-subtrees                         :  cn=monitor
excluded-subtrees                         :  cn=proxy manager
excluded-subtrees                         :  cn=virtual access controls
excluded-subtrees                         :  dc=example,dc=com
filter-join-rule                          :  -
is-enabled                                :  true
is-read-only                              :  false
is-routable                               :  true
ldap-data-source-pool                     :  defaultDataSourcePool
lexicographic-attrs                       :  all
lexicographic-lower-bound                 :  none
lexicographic-upper-bound                 :  none
non-viewable-attr                         :  -
non-writable-attr                         :  -
numeric-attrs                             :  all
numeric-default-data-view                 :  false
numeric-lower-bound                       :  none
numeric-upper-bound                       :  none
pattern-matching-base-object-search-filter :  all
pattern-matching-dn-regular-expression    :  all
pattern-matching-one-level-search-filter  :  all
pattern-matching-subtree-search-filter    :  all
```

```
process-bind                    :  -
replication-role                :  master
viewable-attr                   :  all except non-viewable-attr
writable-attr                   :  all except non-writable-attr
```

Data Views That Route Requests When a List of Subtrees Is Stored on Multiple, Data-Equivalent Data Sources

This section describes how to configure a data view that routes requests targeted at a list of subtrees to a set of data-equivalent data sources. For information about this type of deployment, see "Data Views to Route Requests When a List of Subtrees Are Stored on Multiple, Data-Equivalent Data Sources" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The example in this section has multiple data sources that contain the same set of subtrees. The data sources are data-equivalent and are pooled into one data source pool for load balancing. A data view is configured for each subtree to expose that subtree to client requests. The following figure shows the sample deployment.

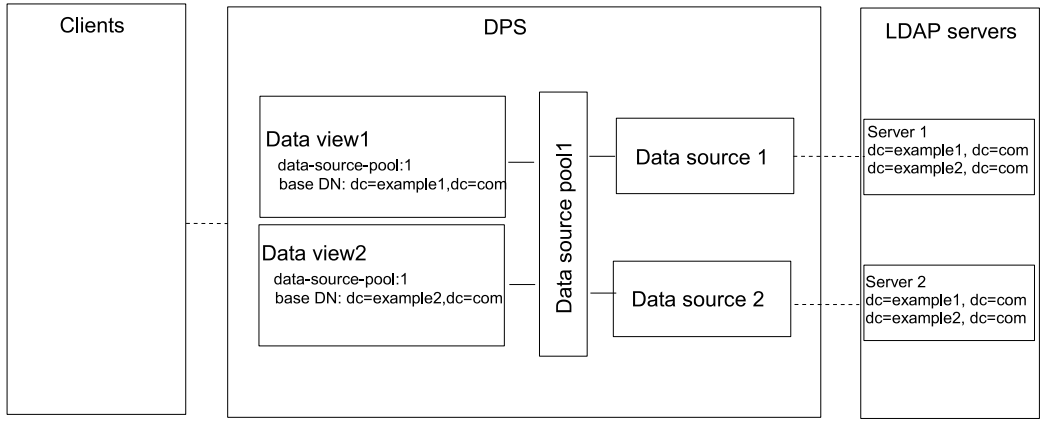


FIGURE 18–1 Sample Deployment That Routes Requests When a List of Subtrees Is Stored on Multiple, Data-Equivalent Data Sources

▼ To Configure Data Views That Route Requests When a List of Subtrees Is Stored on Multiple, Data-Equivalent Data Sources

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Create a data source for each LDAP server as described in "Creating and Configuring LDAP Data Sources" on page 373.**

2. **Create a data source pool as described in "Creating and Configuring LDAP Data Source Pools" on page 376.**

3. **Attach the data sources to the data source pool as described in "Attaching LDAP Data Sources to a Data Source Pool" on page 377.**

4. **(Optional) Configure load balancing.**

   For information, see "Configuring Load Balancing" on page 405.

5. **Create a data view with a base DN at `dc=example1,dc=com` that refers to the data source pool.**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
   data-source-pool-1 dc=example1,dc=com
   ```

6. **Create another data view with a base DN at `dc=example2,dc=com` that refers to the data source pool.**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
   data-source-pool-1 dc=example2,dc=com
   ```

   The other properties of the data views are the same as the default data view in "Default Data View" on page 387.

7. **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

   For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Data Views That Provide a Single Point of Access When Different Subtrees Are Stored in Different Data Sources

This section describes how to configure a data view that provides a single point of access to different subtrees stored in multiple data sources. For information about this type of deployment, see "Data Views to Provide a Single Point of Access When Different Subtrees Are Stored on Different Data Sources" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The example in this section contains a data view for each subtree. A data source pool is configured for each set of data-equivalent data sources. The following figure shows the example deployment.

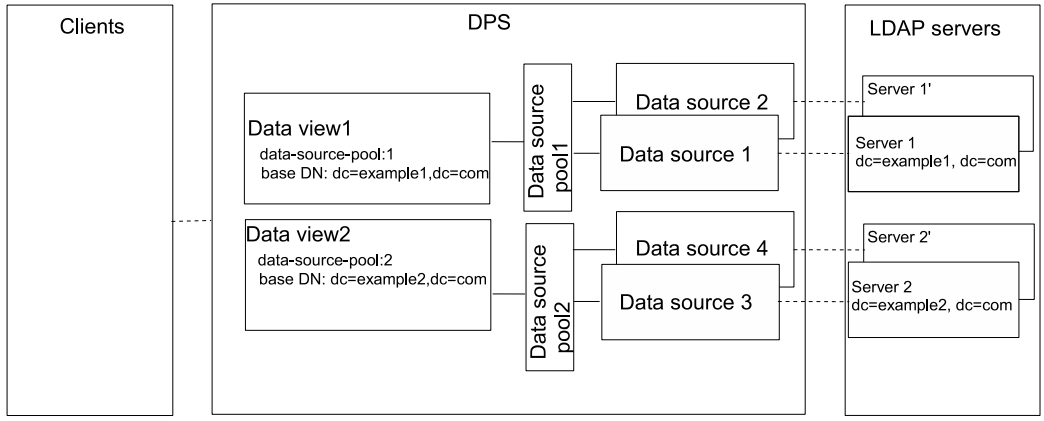


FIGURE 18–2 Sample Deployment That Provides a Single Point of Access When Different Subtrees Are Stored on Different Data Sources

▼ To Configure Data Views That Provide a Single Point of Access When Different Subtrees Are Stored on Different Data Sources

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Create a data source for each LDAP server as described in "Creating and Configuring LDAP Data Sources" on page 373.**

2. **Create two data source pools as described in "Creating and Configuring LDAP Data Source Pools" on page 376.**

3. **Attach the data sources that contain `dc=example1,dc=com` to `data-source-pool-1`, and the data sources that contain `dc=example2,dc=com` to `data-source-pool-2`, as described in "Attaching LDAP Data Sources to a Data Source Pool" on page 377.**

4. **(Optional) Configure load balancing.**

   For information, see "Configuring Load Balancing" on page 405.

5 **Create a data view with a base DN at `dc=example1,dc=com` that refers to `data-source-pool-1`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
data-source-pool-1 dc=example1,dc=com
```

6 **Create another data view with a base DN at `dc=example2,dc=com` that refers to `data-source-pool-2`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
data-source-pool-1 dc=example2,dc=com
```

The other properties of the data views are the same as the default data view in "Default Data View" on page 387.

7 **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Data Views That Provide a Single Point of Access When Superior and Subordinate Subtrees Are Stored in Different Data Sources

This section describes how to configure a data view for a single point of access when a superior branch of a subtree is stored in a different data source to a subordinate branch. For information about this type of deployment, see "Data Views to Route Requests When Superior and Subordinate Subtrees Are Stored in Different Data Sources" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The example in this section contains three data views. The base DN of data view 1 is superior to the base DN of data view 2 and the base DN of data view 3. Or, in other words, data source 2 and data source 3 contain subtrees that are subordinate to the subtree on data source 1. The following figure shows the example deployment.

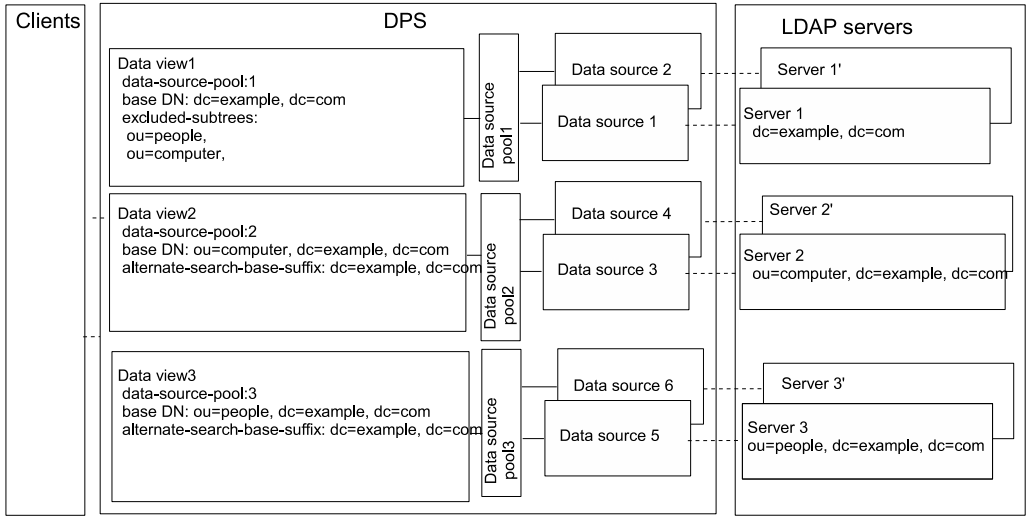


FIGURE 18–3 Sample Deployment to Routes Requests When Superior and Subordinate Subtrees Are Stored in Different Data Sources

Directory Proxy Server automatically excludes a subordinate branch of a subtree from a data view when the subordinate branch is configured as the base DN of a separate data view.

▼ To Configure Data Views That Provide a Single Point of Access When Superior and Subordinate Subtrees Are Stored in Different Data Sources

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Create a data source for each LDAP server as described in "Creating and Configuring LDAP Data Sources" on page 373.**

2. **Create three data source pools as described in "Creating and Configuring LDAP Data Source Pools" on page 376.**

3. **Attach the data sources to the data source pools by following the instructions in "Attaching LDAP Data Sources to a Data Source Pool" on page 377.**

   - Attach the data sources that contain `dc=example,dc=com` to `data-source-pool-1`.
   - Attach the data sources that contain `ou=computer,dc=example,dc=com` to `data-source-pool-2`.

- Attach the data sources that contain `ou=people,dc=example,dc=com` to `data-source-pool-3`.

4 **(Optional) Configure load balancing.**

For information, see "Configuring Load Balancing" on page 405.

5 **Create a data view with a base DN at `dc=example,dc=com` and a data source pool `data-source-pool-1`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
 data-source-pool-1 dc=example,dc=com
```

6 **Create a data view with a base DN at `ou=computer,dc=example,dc=com` and a data source pool `data-source-pool-2`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
 data-source-pool-2 ou=computer,dc=example,dc=com
```

7 **Create a data view with a base DN at `ou=people,dc=example,dc=com` and a data source pool `data-source-pool-3`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
 data-source-pool-3 ou=people,dc=example,dc=com
```

8 **Verify that the subtrees `ou=computer,dc=example, dc=com` and `ou=people,dc=example, dc=com` have been excluded from `dataview-1` by looking at the `excluded-subtrees` parameter.**

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
```

The list of excluded subtrees is returned.

9 **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

◆ ◆ ◆ CHAPTER 19

Directory Proxy Server Certificates

This chapter describes how to configure certificates on Directory Proxy Server. For information about configuring certificates on *Directory Server*, see "Managing Certificates" on page 103.

The procedures in this chapter use the `dpadm` and `dpconf` commands. For information about these commands, see the `dpadm(1M)` and `dpconf(1M)` man pages.

This chapter covers the following topics:

Default Self-Signed Certificate

When you create a Directory Proxy Server instance, it has a default self-signed certificate. A self-signed certificate is a public and private key pair, where the public key is self-signed by Directory Proxy Server.

▼ Viewing the Default Self-Signed Certificate

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **View the default self-signed certificate.**

  ```
  $ dpadm show-cert instance-path defaultservercert
  ```

Creating, Requesting and Installing Certificates for Directory Proxy Server

To run the Secure Sockets Layer (SSL) on Directory Proxy Server, you must either use a self-signed certificate or a Public Key Infrastructure (PKI) solution.

The PKI solution involves an external Certificate Authority (CA). For a PKI solution you need a CA-signed server certificate, which contains both a public key and a private key. This certificate is specific to one Directory Proxy Server instance. You also need a *trusted CA certificate*, which contains a public key. The trusted CA certificate ensures that all server certificates from your CA are trusted. This certificate is sometimes called a CA root key or root certificate.

For information about how to create a non-default self-signed certificate and to request and install a CA-signed certificate, see the following procedures.

▼ To Create a Non-default Self-Signed Certificate for Directory Proxy Server

When you create a Directory Proxy Server instance, a default self-signed certificate is automatically provided. If you want to create a self-signed certificate with non-default settings, use this procedure.

The procedure creates the public and private key pair for a server certificate, where the public key is signed by Directory Proxy Server. A self-signed certificate is valid for three months.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **To create a non-default self-signed certificate for Directory Proxy Server, type:**

  ```
  $ dpadm add-selfsign-cert instance-path cert-alias
  ```

  where *cert-alias* is the name of the self-signed certificate.

  For example, you could create a certificate called `my-self-signed-cert` as follows:

  ```
  $ dpadm add-selfsign-cert /local/dps my-self-signed-cert
  ```

  For a description of all command options, see the `dpadm(1M)` man page or type `dpadm add-selfsign-cert --help` at the command line.

▼ To Request a CA-Signed Certificate for Directory Proxy Server

Self-signed certificates are useful for test purposes. However, in a production environment, using trusted Certificate Authority (CA) certificates is more secure.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Request a CA-signed server certificate.

```
$ dpadm request-cert instance-path cert-alias
```

where *cert-alias* is the name of the certificate that you are requesting. Certificate Authorities might require all of the options of the command to identify the server. For a description of all command options, see the `dpadm(1M)` man page.

The process for obtaining a CA certificate depends on the CA that you use. Some commercial CAs provide a web site that allows you to download the certificate. Other CAs will send the certificate to you in email.

For example, you could request a certificate called `my-CA-signed-cert` as follows:

```
$ dpadm request-cert -S cn=my-request,o=test /local/dps my-CA-signed-cert
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBYDCBygIBADAhMQ0wCwYDVQQDEwRnZXJpMRAwDgYDVQQDEwdteWNlcnQ0MIGfMA0GCSqGSIb3
DQEBAQUAA4GNADCBiQKBgQC3v9ubG468wnjBDAMbRrEkmFDTQzT+LO30D/ALLXOiElVsHrtRyWhJ
PG9cURI9uwqs15crxCpJvho1kt3SB9+yMB8Ql+CKnCQDHlNAfnn30MjFHShv/sAuEygFsN+Ekci5
W1jySYE2rzE0qKVxWLSILFo1UFRVRsUnORTX/Nas7QIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEA
fcQMnZNLpPobiX1xy1ROefPOhksVz8didY8Q2fjjaHG5lajMsqORO zubsuQ9Xh4ohT8kIA6xcBNZ
g8FRNIRAHCtDXKOdOm3CpJ8da+YGI/ttSawIeNAKU1DApF9zMb7c2lS4yEfWmreoQdXIC9YeKtF6
zwbn2EmIpjHzETtS5Nk=
-----END NEW CERTIFICATE REQUEST-----
```

When you request a certificate by using the `dpadm request-cert` command, the certificate request is a PKCS #10 certificate request in Privacy Enhanced Mail (PEM) format. PEM is the format specified by RFCs 1421 through 1424. For more information, see `http://www.ietf.org/rfc/rfc1421.txt`. The PEM format represents a base64-encoded certificate request in ASCII format.

When you request a CA-signed certificate, a temporary self-signed certificate is created. When you receive and install the CA-signed certificate from the CA, the new certificate replaces the temporary self-signed certificate.

Save the certificate request for future reference. You may need it while renewing the certificate.

2 **Send the certificate request to the CA, according to its procedures.**

After you have sent your request, you must wait for the CA to respond with your certificate. Response time for your request varies. For example, if your CA is internal to your company, the response time can be short. However, if the CA is external to your company, the CA can take several weeks to respond to your request.

3 **Save the certificate that you receive from the CA.**

Save your certificate in a text file, and back up the certificate in a safe location.

▼ To Install a CA-Signed Server Certificate for Directory Proxy Server

To trust the CA-signed server certificate, you must install the certificate on a Directory Proxy Server instance. This procedure installs the public key of a CA certificate to the certificate database on Directory Proxy Server.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **See if the trusted CA certificate for this CA is already installed.**

To do this, list all installed CA certificates, as described in "To List CA Certificates" on page 400.

2 **If the trusted CA certificate is not installed, add it to the certificate database on the Directory Proxy Server instance.**

```
$ dpadm add-cert instance-path cert-alias cert-file
```

where *cert-alias* is the name of the trusted CA certificate and *cert-file* is the name of the file containing the trusted CA certificate.

3 **Install the CA-signed server certificate to the certificate database.**

```
$ dpadm add-cert instance-path cert-alias cert-file
```

Where *cert-alias* is the name of the CA-signed server certificate and *cert-file* is the name of the file containing the CA-signed server certificate.

Note – The *cert-alias* name must be the same as the *cert-alias* used in the certificate request.

For example, you can add a CA-signed server certificate named `CA-cert` to the certificate database on `/local/dps` as follows:

```
$ dpadm add-cert /local/dps CA-cert /local/safeplace/ca-cert-file.ascii
```

Renewing an Expired CA-Signed Certificate for Directory Proxy Server

This section describes how to renew an expired CA-signed server certificate.

▼ To Renew an Expired CA-Signed Server Certificate for Directory Proxy Server

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Obtain an updated certificate from your CA.**

2 **Install the certificate on your instance of Directory Proxy Server.**

```
$ dpadm renew-cert instance-path cert-alias cert-file
```

where *cert-alias* is the name of the new certificate and *cert-file* is the name of the file containing the certificate. For a description of all command options, see the `dpadm(1M)` man page.

Listing Certificates

For information about how to list server and CA certificates, see the following procedures.

▼ To List Server Certificates

This procedure lists all certificates that are installed on an instance of Directory Proxy Server.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **List the server certificates in the certificate database on the Directory Proxy Server instance.**

```
$ dpadm list-certs instance-path
```

By default, an instance of Directory Proxy Server contains a server certificate named `defaultservercert`. The text `Same as issuer` indicates that the default certificate is a self-signed server certificate.

For example:

```
$ dpadm list-certs /local/dps
Alias            Valid from       Expires on       Self-signed? Issued by         Issued to
```

```
---------------- ---------------- ---------------- ------------ ------------------ -------------
defaultservercert 2006/06/01 04:15 2008/05/31 04:15 y            CN=myserver:myport Same as issuer
1 certificate found.
```

▼ To List CA Certificates

This procedure lists CA certificates that are installed on an instance of Directory Proxy Server.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **List the CA certificates in the certificate database on the Directory Proxy Server instance.**

  `$ dpadm list-certs -C` *instance-path*

  For example:

```
$ dpadm list-certs -C /local/dps
Alias   Valid from       Expires on        Built-in  Issued by    Issued to
------  ----------       ----------------  --------- ---------    ---------
CAcert1 1999/06/21 06:00 2020/06/21 06:00  y         CN=company1, O=company2
...
```

Adding a Certificate From a Back-End LDAP Server to the Certificate Database on Directory Proxy Server

This section describes how to add a certificate from a back-end LDAP server to the certificate database on Directory Proxy Server.

▼ To Add a Certificate From a Back-End Directory Server to the Certificate Database on Directory Proxy Server

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Display the certificate from the back-end Directory Server in PEM format by using this command syntax:**

   `dsadm show-cert -F ascii` *instance-path* `[`*cert-alias*`]`

   If you do not specify a *cert-alias*, the default server certificate is displayed. For a description of all command options, see the `dsadm`(1M) man page.

For example, show the default self-signed server certificate as follows:

```
$ dsadm show-cert -F ascii /local/dsInst defaultCert
-----BEGIN CERTIFICATE-----
MIICJjCCAY+gAwIBAgIFAIKL36kwDQYJKoZIhvcNAQEEBQAwVzEZMBcGA1UEChMQ
U3VuIE1pY3Jvc3lzdGVtczEZMBcGA1UEAxMQRGlyZWN0b3J5IFNlcnZlcjENMAsG
A1UEAxMEMjAxMTEQMA4GA1UEAxMHY29uZHlsZTAeFw0wNjA1MjIxMTQxNTVaFw0w
NjA4MjIxMTQxNTVaMFcxGTAXBgNVBAoTEFN1biBNaWNyb3N5c3RlbXMxGTAXBgNV
BAMTEERpcmVjdG9yeSBTZXJ2ZXIxDTALBgNVBAMTBDIwMTExEDAOBgNVBAMTB2Nv
bmR5bGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK9U3ry3sJmEzwQY8CGd
7S2MTZuBedo03Vea1lfDtD08WIsdDMzhHplTdeHAkWWNc8g2PDcEFXeWp9UXFMuD
Pcia7t8HtFkm73VmlriWhMd8nn3l2vkxhsPK2LHFEeOIUDR9LBBiMiEeLkjdoEhE
VLMSoYKqKI+Aa5grINdmtFzBAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAF4eDbSd7
qy2l10dIogT+rnXZ362gLTlQFCblhbGpmmptbegUdL1ITGv/62q1isPV2rW7CkjM
Cqb0fo3k5UkKKvW+JbMowpQeAPnlgpX612HuDr1tldnKV4eyU7gpG31t/cpACALQ
7OPi1A7oVb2Z8OJKfEJHkp3txBSsiI2gTkk=
-----END CERTIFICATE-----
```

2 Save the certificate.

Save your certificate in a text file, and back up the certificate in a safe location.

3 Add the certificate from the back-end LDAP server to the certificate database on an instance of Directory Proxy Server.

```
$ dpadm add-cert instance-path cert-alias cert-file
```

where *cert-alias* is the name of the certificate and *cert-file* is the name of the file containing the certificate.

For example, you could add the certificate `defaultCert` as follows:

```
$ dpadm add-cert /local/dps defaultCert /local/safeplace/defaultCert.ascii
```

Exporting a Certificate to a Back-End LDAP Server

Back-end LDAP servers might require a certificate from Directory Proxy Server. This section describes how to configure Directory Proxy Server to export a certificate to a back-end LDAP server.

▼ To Configure Directory Proxy Server to Export a Client Certificate to a Back-End LDAP Server

1 Specify the certificate to be sent to the back-end LDAP server.

```
$ dpconf set-server-prop -h host -p port ssl-client-cert-alias:cert-alias
```

Where *cert-alias* is the name of the certificate. For a description of all command options, see the `dpconf(1M)` man page.

2 **Copy the contents of the certificate to a file.**

```
$ dpadm show-cert -F ascii -o filename instance-path cert-alias
```

3 **Add the certificate to the certificate database for the back-end LDAP server as described in "To Add the CA-Signed Server Certificate and the Trusted CA Certificate" on page 106.**

Next Steps Configure the back-end LDAP server for client authentication. For information about how to do this for Directory Server, see "Configuring Credential Levels and Authentication Methods" on page 113.

See Also For information about configuring certificate-based authentication between clients and Directory Proxy Server, see "To Configure Certificate-based Authentication" on page 509.

Backing Up and Restoring a Certificate Database for Directory Proxy Server

Server certificates are backed up when you use `dpadm` to back up Directory Proxy Server. The backed up certificates are stored in the *archive-path*`/alias` directory.

For information about how to back up and restore Directory Proxy Server, see "Backing Up and Restoring Directory Proxy Server Instances" on page 371.

Prompting for a Password to Access the Certificate Database

By default, the password for the certificate database is managed internally. Therefore, you do not need to type a certificate password or specify the password file. When the certificate database is managed internally through a stored password, the password is stored in a secure environment.

For more security and more control over certificates, configure Directory Proxy Server to prompt for a password on the command line. You are then prompted to enter the password for all `dpadm` subcommands except `autostart`, `backup`, `disable-service`, `enable-service`, `info`, `restore`, and `stop`.

For information about configuring Directory Proxy Server to prompt or not to prompt for passwords, see the following procedures.

▼ To Prompt for a Password to Access the Certificate Database

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Stop the server.**

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2 **Set the password prompt flag to `on`, then type and confirm the certificate database password.**

```
$ dpadm set-flags instance-path cert-pwd-prompt=on
Choose the certificate database password:
Confirm the certificate database password:
```

3 **Start the server, then type the certificate database password.**

```
$ dpadm start instance-path
Enter the certificate database password:
```

▼ To Disable the Password Prompt to Access the Certificate Database

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Stop the server.**

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2 **Set the password prompt flag to `off`, then type the existing password.**

```
$ dpadm set-flags instance-path cert-pwd-prompt=off
Enter the old password:
```

3 **Start the server.**

```
$ dpadm start instance-path
```

◆ ◆ ◆ CHAPTER 20

Directory Proxy Server Load Balancing and Client Affinity

For a description of load balancing and client affinity, see Chapter 16, "Directory Proxy Server Load Balancing and Client Affinity," in *Sun Directory Server Enterprise Edition 7.0 Reference*. This chapter covers the following topics:

Configuring Load Balancing

For information about load balancing, see "Load Balancing" in *Sun Directory Server Enterprise Edition 7.0 Reference*. This section explains how to configure load balancing and provides sample configurations.

▼ To Select a Load Balancing Algorithm

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Obtain the current load balancing algorithm by viewing the properties of the LDAP data source pool.

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

The default properties of an LDAP data source pool are as follows:

```
client-affinity-bind-dn-filters     :  any
client-affinity-criteria            :  connection
client-affinity-ip-address-filters  :  any
client-affinity-policy              :  write-affinity-after-write
client-affinity-timeout             :  20s
description                         :  -
```

```
enable-client-affinity              :  false
load-balancing-algorithm            :  proportional
```

By default, the load balancing algorithm is `proportional`.

2 Configure the LDAP data source pool to use an algorithm.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:selected-algorithm
```

where *selected-algorithm* is one of the following:

- `failover`
- `operational-affinity`
- `proportional`
- `saturation`

For more information about the algorithms, see "Introduction to Load Balancing" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

3 Restart the instance of Directory Proxy Server.

```
$ dpadm restart instance-path
```

▼ To Configure Weights for Load Balancing

You need to configure the weights of an attached data source in relation to the weights of any other attached data sources in the data source pool. Consider the weights of all of your attached data sources. If a data source has a weight of `disabled` for a type of operation, requests of that type are never sent to that data source. If a data source has a weight of `0` (zero), no requests are distributed to that data source.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the list of data sources that are attached to the data source pool.

```
$ dpconf list-attached-ldap-data-sources -h host -p port pool-name
```

2 View the properties of one of the attached data sources.

```
$ dpconf get-attached-ldap-data-source-prop pool-name \
 attached-data-source-name
```

The properties of an attached data source define the weight for each type of operation. The default weights of an attached data source are as follows:

```
add-weight          :  disabled
bind-weight         :  disabled
compare-weight      :  disabled
```

```
delete-weight     :  disabled
modify-dn-weight  :  disabled
modify-weight     :  disabled
search-weight     :  disabled
```

3 **Configure the weights of one of the attached data sources.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name \
 attached-data-source-name add-weight:value \
 bind-weight:value compare-weight:value delete-weight:value \
 modify-dn-weight:value modify-weight:value search-weight:value
```

4 **Repeat Step 2 and Step 3 for the other attached data sources.**

5 **Compare the key parameters of the attached data sources.**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

For example, a data source pool can contain data sources with the following weights:

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v myPool
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
DS-1     disabled   3               disabled       disabled
DS-2     2          2           2              2
DS-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
disabled         disabled      disabled
2                2             2
1                1             1
```

Example Configurations for Load Balancing

This section contains sample procedures for configuring each of the load balancing algorithms.

▼ To Configure the Proportional Algorithm for Load Balancing

For a description of the proportional algorithm, see "Proportional Algorithm for Load Balancing" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

In this example, the data source *ds–1* is configured with twice the weight of the other two data sources.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Ensure that you have a data source pool with at least three attached data sources. For information about how to create data sources and data source pools, see "Creating LDAP Data Views" on page 373.

1 Configure the data source pool to use the proportional algorithm for load balancing.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:proportional
```

2 Configure the properties of the first data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

3 Configure the properties of the second data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 Configure the properties of the third data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 Compare the key parameters of the attached data sources.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     2          2           2              2
ds-2     1          1           1              1
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
2                2             2
1                1             1
1                1             1
```

6 Restart the instance of Directory Proxy Server.

```
$ dpadm restart instance-path
```

▼ To Configure the Saturation Algorithm for Load Balancing

For a description of the saturation algorithm, see "Saturation Algorithm for Load Balancing" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

In this example, the data source `ds-1` performs the majority of bind operations but does not perform any other types of operations. The three data sources are configured with the following weights :

- `ds-1` is configured with weight 3 for bind operations and is disabled for all other types of operations.
- `ds-2` is configured with weight 2 for all operations.
- `ds-3` is configured with weight 1 for all operations.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Ensure that you have a data source pool with at least three attached data sources. For information about how to create data sources and data source pools, see "Creating LDAP Data Views" on page 373.

1 Configure the data source pool to use the saturation algorithm for load balancing.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:saturation
```

2 Configure the properties of the first data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:disabled bind-weight:3 compare-weight:disabled delete-weight:disabled \
 modify-dn-weight:disabled modify-weight:disabled search-weight:disabled
```

3 Configure the properties of the second data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4 Configure the properties of the third data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 Compare the key parameters of the attached data sources.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     disabled   3               disabled       disabled
ds-2     2          2           2              2
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
```

```
---------------- ------------- -------------
disabled         disabled      disabled
2                2             2
1                1             1
```

6 **Restart the instance of Directory Proxy Server.**

```
$ dpadm restart instance-path
```

▼ To Configure the Operational Affinity Algorithm for Global Account Lockout

For a description of this algorithm, "Operational Affinity Algorithm for Global Account Lockout" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This example has three data sources. The data source `ds-1` is configured to receive all bind requests.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Ensure that you have a data source pool with at least three attached data sources. For information about how to create data sources and data source pools, see "Creating LDAP Data Views" on page 373.

1 **Configure the data source pool to use the operational affinity algorithm.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2 **Configure the properties of the first data source.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:100 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

3 **Configure the properties of the second data source.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 **Configure the properties of the third data source.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 **Compare the key parameters of the attached data sources.**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     1          100         1              1
ds-2     1          1           1              1
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
1                1             1
1                1             1
1                1             1
```

6 **Restart the instance of Directory Proxy Server.**

```
$ dpadm restart instance-path
```

▼ To Configure Operational Affinity Algorithm for Cache Optimization

For a description of this algorithm, see "Operational Affinity Algorithm for Cache Optimization" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This example has three data sources. All search and compare operations are treated by the data source `ds-1`. When `ds-1` responds to a request, the targeted entry is stored in the cache. If `ds-1` responds repeatedly to the same request, the data source can use cached data.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Ensure that you have a data source pool with at least three attached data sources. For information about how to create data sources and data source pools, see "Creating LDAP Data Views" on page 373.

1 **Configure the data source pool to use the operational affinity algorithm.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2 **Configure the properties of the first data source.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:1 compare-weight:100 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:100
```

3 **Configure the properties of the second data source.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 Configure the properties of the third data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 Compare the key parameters of the attached data sources.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     1          1           100            1
ds-2     1          1           1              1
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
1                1             100
1                1             1
1                1             1
```

6 Restart the instance of Directory Proxy Server.

```
$ dpadm restart instance-path
```

▼ To Configure the Failover Algorithm for Load Balancing

For a description of the failover algorithm, see "Failover Algorithm for Load Balancing" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This example has three data sources. The data source `ds-1` receives all requests. If `ds-1` fails, `ds-2` receives all requests until `ds-1` recovers. If `ds-2` fails before `ds-1` recovers, `ds-3` receives all requests.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Ensure that you have a data source pool with at least three attached data sources. For information about how to create data sources and data source pools, see "Creating LDAP Data Views" on page 373.

1 Configure the data source pool to use the failover algorithm for load balancing.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:failover
```

2 Configure the properties of the first data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:3 bind-weight:3 compare-weight:3 delete-weight:3 modify-dn-weight:3 \
 modify-weight:3 search-weight:3
```

3 Configure the properties of the second data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4 Configure the properties of the third data source.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 Compare the key parameters of the attached data sources.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     3          3           3              3
ds-2     2          2           2              2
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
3                3             3
2                2             2
1                1             1
```

6 Restart the instance of Directory Proxy Server.

```
$ dpadm restart instance-path
```

Configuring Directory Proxy Server To Perform Load Balancing

A simple case of load balancing consists of sending search and compare operations to one set of directories, and sending other operations to another set. Directory Proxy Server receives all client operations. The server must determine which set gets the reads, and which set gets the other operations.

The key stages in configuring Directory Proxy Server to handle this load balancing scenario are as follows.

1. Add directories as data sources for Directory Proxy Server.
2. Add the data sources to a data source pool.
3. Configure some of the data sources to accept search and compare, other data sources to accept add, bind, delete, modify, and modify DN operations.
4. Add the data source pool to a data view.

The following example involves Directory Proxy Server, listening on port 9389. The proxy is configured here to balance the load as described across one Directory Server instance, `ds1:1389`, handling search and compare operations, and another Directory Server instance, `ds2:2389`, handling other operations.

The first step creates the data sources, and enables the data sources. This step requires a proxy server restart.

```
$ dpconf create-ldap-data-source -p 9389 ds1 localhost:1389
$ dpconf create-ldap-data-source -p 9389 ds2 localhost:2389
$ dpconf set-ldap-data-source-prop -p 9389 ds1 is-enabled:true
$ dpconf set-ldap-data-source-prop -p 9389 ds2 is-enabled:true
$ dpadm restart /local/dps
```

The second step adds the data sources to a data source pool.

```
$ dpconf create-ldap-data-source-pool -p 9389 "Directory Pool"
$ dpconf attach-ldap-data-source -p 9389 "Directory Pool" ds1 ds2
```

The third step configures `ds1` to accept search and compare operations, `ds2` to accept other operations.

```
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Directory Pool" ds1 \
add-weight:disabled bind-weight:disabled compare-weight:1 delete-weight:disabled \
modify-dn-weight:disabled modify-weight:disabled search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Directory Pool" ds2 \
add-weight:1 bind-weight:1 compare-weight:disabled delete-weight:1 \
modify-dn-weight:1 modify-weight:1 search-weight:disabled
```

The fourth step adds the data source pool to a data view, so that client application requests are routed to the pool.

```
$ dpconf create-ldap-data-view -p 9389 "Balanced View" "Directory Pool" \
dc=example,dc=com
```

Configuring Client Affinity

Client affinity reduces the risk of propagation delay in load-balanced deployments. For information about client affinity, see "Client Affinity" in *Sun Directory Server Enterprise Edition 7.0 Reference*. This section explains how to configure affinity between a client connection and a data source, and provides sample configurations.

▼ To Configure Client Affinity

This procedure describes how to configure affinity between a client connection and a data source.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the current load balancing algorithm by viewing the properties of the data source pool.

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

The default properties of a data source pool are as follows:

```
client-affinity-bind-dn-filters     :  any
client-affinity-criteria            :  connection
client-affinity-ip-address-filters  :  any
client-affinity-policy              :  write-affinity-after-write
client-affinity-timeout             :  20s
description                         :  -
enable-client-affinity              :  false
load-balancing-algorithm            :  proportional
```

These parameters configure client affinity: `client-affinity-bind-dn-filters`, `client-affinity-criteria`, `client-affinity-ip-address-filters`, `client-affinity-policy`, `client-affinity-timeout`, and `enable-client-affinity`. For a description of the properties and a list of their valid values, type:

```
dpconf help-properties ldap-data-source-pool client-affinity-bind-dn-filters \
client-affinity-criteria client-affinity-policy client-affinity-ip-address-filters\
 client-affinity-timeout enable-client-affinity
```

For more information about the properties, see these man pages: `client-affinity-bind-dn-filters`(5dpconf), `client-affinity-criteria`(5dpconf), `client-affinity-ip-address-filters`(5dpconf)`client-affinity-policy`(5dpconf), `client-affinity-timeout`(5dpconf), and `enable-client-affinity`(5dpconf).

2 Enable client affinity.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 enable-client-affinity:true
```

3 Select a policy for client affinity.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:selected-policy
```

where *selected-policy* is one of the following:

`write-affinity-after-write`
: Affinity for write requests after the first write request

`read-write-affinity-after-write`
: Affinity for all requests after the first write request

`read-write-affinity-after-any`
: Affinity for all requests after the first read request or write request

`read-affinity-after-write`
Affinity for the first read request after a write request

4 **Configure the duration of the client affinity.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-timeout:time-out[unit]
```

The default *unit* for timeout is milliseconds.

The above setting is applicable to the only connection under consideration. It is not applicable to all the connections from a particular client.

Example Configurations for Client Affinity

This section contains example configurations related to client affinity, and includes examples for replication delay, verifying write operations, and connection-based routing.

▼ To Configure Client Affinity for Replication Delay When a Data Source Pool Contains Masters and Consumers

This procedure configures client affinity for all read and write operations that occur up to three seconds after the first write operation.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure the affinity parameters for the data source pool.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-write client-affinity-timeout:3000 \
 enable-client-affinity:true
```

▼ To Configure Client Affinity to Verify Each Write Operation With a Read Operation

This procedure configures client affinity for the first read operation after each write operation. The example could be for an application where a specified bind DN validates each write operation by performing a read operation.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure the affinity parameters for the data source pool.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-affinity-after-write enable-client-affinity:true
```

To Configure Client Affinity for Client—Based Routing

If an application makes an update using one connection from the pool but then uses a different connection to do the search for that entry, the affinity setting on the connection used to do the update is not used because the search is done from a different connection. The search operation could also be routed to a different server than where the update was performed. In this case, the affinity feature works only within the same client connection.

To resolve this, affinity should be defined at the client level such as an IP address or bind DN. When an update is made by a client, all the connections from that client follow the same affinity rule.

1. **Specify the criteria to determine if the requests are coming from the same client.**

   ```
   dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
   client-affinity-criteria:ip-address-and-bind-dn
   ```

   For all the options, see `client-affinity-criteria(5dpconf)`.

   The server matches the bind DN as well as the IP address of the client requests, if the entries meet the criteria then they are from the same client.

2. **Specify the regular expressions that the bind DN of the connection must match to consider that requests come from the same client.**

   ```
   dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
   client-affinity-bind-dn-filters:"uid=boss*"
   ```

3. **Specify the IPv4 or IPv6 address that the IP address of the connection must match to consider that requests come from the same client.**

   ```
   dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
   client-affinity-ip-address-filters:129.157.192.108
   ```

To Configure Client Affinity for Connection-Based Routing

In versions prior to Directory Proxy Server 6.0, one connection was opened between a client and an LDAP server. The same connection was used for all requests from the client until the connection was closed. This type of routing is called *connection-based routing*. This procedure describes how to configure client affinity for connection-based routing.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Before You Begin Ensure that all data sources are attached to the data source pool and that `client-cred-mode` is set to `use-client-identity`.

- **Configure the affinity parameters for the data source pool.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-any enable-client-affinity:true
```

◆ ◆ ◆ CHAPTER 21

Directory Proxy Server Distribution

Directory Proxy Server enables distribution through the definition of data views. Data views are defined with a view base, which determines the base DN of the entries in that data view. Based on the distribution algorithms provided in Directory Proxy Server, you can specify how entries are divided among the different data views.

For an overview of Directory Proxy Server distribution and a description of example use cases, see Chapter 17, "Directory Proxy Server Distribution," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Configuring Directory Proxy Server Distribution Algorithms

Directory Proxy Server provides the following distribution algorithms:

- Pattern matching
- Numeric
- Lexicographic
- Replication
- Custom

Configuring Pattern Matching Distribution Algorithm

Directory Proxy Server distributes the requests to data views based on the match between the parameters of the requests and one or more patterns. Set the following parameters to configure the Pattern matching distribution algorithm:

- `pattern-matching-base-dn-regular-expression`
  `pattern-matching-base-dn-regular-expression(5dpconf)`
- `pattern-matching-base-object-search-filter`
  `pattern-matching-base-object-search-filter(5dpconf)`
- `pattern-matching-dn-regular-expression`
  `pattern-matching-dn-regular-expression(5dpconf)`
- `pattern-matching-one-level-search-filter`
  `pattern-matching-one-level-search-filter(5dpconf)`
- `pattern-matching-subtree-search-filter`
  `pattern-matching-subtree-search-filter(5dpconf)`

All the pattern matching distribution algorithm properties are multivalued. Use `PROP+:VAL` to add a value, and `PROP-:VAL` to remove a value. For example:

```
$ dpconf set-ldap-data-view-prop -p port-number ldap-data-view \
pattern-matching-dn-regular-expression:value
$ dpconf set-ldap-data-view-prop -p port-number ldap-data-view \
pattern-matching-dn-regular-expression+:value2
```

The order in which values are set, the priority is decided.

```
$ ldapsearch -D "cn=proxy manager" -w - -p port-number -b "cn=ldap-data-view,cn=data views,cn=config" \
"objectclass=*" dnMatchingRegex

version: 1
dn: cn=ldap-data-view,cn=data views,cn=config
dnMatchingRegex: 1:value
dnMatchingRegex: 2:value2
```

In the above example, the value prefixed by `1` is of highest priority.

To switch back to version 6 behavior for pattern matching distribution algorithm, set the `compat-flag` Directory Proxy Server configuration property to `pattern-matching-algo-6`.

The configuration attributes that end with `filter` are LDAP filters, not regular expressions. These LDAP filters are evaluated against LDAP filters contained in the incoming search requests.

For example, use the following settings to configure the Pattern Matching distribution algorithm to send the requests for the users with even `uid` to `even` data view and the users with odd `uid` to odd data view.

```
$ dpconf set-ldap-data-view-prop even
pattern-matching-base-object-search-filter:'|(uid=\2a)(uid=*0)(uid=*2)\
(uid=*4)(uid=*6)(uid=*8))'\
pattern-matching-one-level-search-filter:'|(uid=\2a)(uid=*0)(uid=*2)\
```

```
(uid=*4)(uid=*6)(uid=*8))'\
pattern-matching-subtree-search-filter:'|(uid=\2a)(uid=*0)(uid=*2)\
(uid=*4)(uid=*6)(uid=*8))'\
pattern-matching-dn-regular-expression:'uid=[0-9]+[02468]'
distribution-algorithm: pattern-matching

$ dpconf set-ldap-data-view-prop odd
pattern-matching-base-object-search-filter:'|(uid=\2a)(uid=*1)(uid=*3)\
(uid=*5)(uid=*7)(uid=*9))'\
pattern-matching-one-level-search-filter:'|(uid=\2a)(uid=*1)(uid=*3)\
(uid=*5)(uid=*7)(uid=*9))'\
pattern-matching-subtree-search-filter:'|(uid=\2a)(uid=*1)(uid=*3)\
(uid=*5)(uid=*7)(uid=*9))'\
pattern-matching-dn-regular-expression:'uid=[0-9]+[13579]'
distribution-algorithm: pattern-matching
```

In the `(uid=\2a)` expression, the `\2a` is an ASCII representation of * where `2` and `a` are two hexadecimal digits. The `(uid=\2a)` expression makes sure that the data view accepts the requests for all `uid`s.

The syntax supported by the pattern matching algorithm is specified by the Java Pattern class (documented at (`http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html`)). This syntax is not the same as the usual `regex` syntax.

Configuring Numeric Distribution Algorithm

Directory Proxy Server distributes the requests to data views according to the numeric value of the RDN in the request. The numeric value is taken from the value of the first RDN beneath the base DN of the data view. Set the following parameters define the Numeric bounds:

- `numeric-attrs` `numeric-attrs(5dpconf)`
- `numeric-default-data-view` `numeric-default-data-view(5dpconf)`
- `numeric-lower-bound` `numeric-lower-bound(5dpconf)`
- `numeric-upper-bound` `numeric-upper-bound(5dpconf)`

For example, to configure the numeric distribution algorithm to send the requests for `uid` between 0 to 99 to a specific data view. Use the same syntax for the rest of the users but with a different data view.

```
$ dpconf set-ldap-data-view-prop dataview distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:0 numeric-upper-bound:99
```

Configuring Lexicographic Distribution Algorithm

Directory Proxy Server distributes the requests to data views according to the lexicographic value of the RDN in the request. Lexicographic bounds are taken from the value of the first RDN beneath the base DN of the data view. Set the following parameters to define the Lexicographic bounds:

- `lexicographic-attrs` `lexicographic-attrs(5dpconf)`
- `lexicographic-lower-bound` `lexicographic-lower-bound(5dpconf)`
- `lexicographic-upper-bound` `lexicographic-upper-bound(5dpconf)`

For example, to configure the Lexicographic distribution algorithm to send the requests of the users whose name starts between `A` to `M` to one data view and the requests for the rest of the users to another data view.

```
$ dpconf set-ldap-data-view-prop dataview distribution-algorithm:lexicographic \
 lexicographic-attrs:cn lexicographic-lower-bound:A lexicographic-upper-bound:M
```

Configuring Replication Distribution Algorithm

Directory Proxy Server distributes the requests to data views according to the role of the data view in replication. The algorithm distributes write operations to all data sources in the data source pool and read operations to a single data source. The replication role is defined by the `replication-role` parameter. A data view can have a master role or a consumer role.

```
$ dpconf set-ldap-data-view-prop dataview distribution-algorithm:replication
```

Configuring Custom Distribution Algorithm

Custom distribution algorithm can be configured for all types of data views, that is, `ldap-data-view`, `jdbc-data-view`, `ldif-data-view`, and `join-data-view`. In the following procedure the algorithm is set only for `ldap-data-view`.

▼ To Configure Custom Distribution Algorithm

1. **Set the `extension-jar-file-url` property to contain the path of the Java Archive (JAR) file containing your distribution algorithm class.**

   ```
   $ dpconf set-server-prop -h host -p port extension-jar-file-url:jar file path
   ```

   The *jar file path* can be replaced with a valid JAR file path such as `file:/expt/dps/custom_plugin/myjar.jar`.

2 **Before you configure** `custom-distribution-algorithm`**, set** `distribution-algorithm` **to** `none`**.**

```
$ dpconf set-ldap-data-view-prop view name distribution-algorithm:none
```

3 **Set the** `custom-distribution-algorithm` **property to your custom distribution algorithm class.**

```
$ dpconf set-ldap-data-view-prop view name \
custom-distribution-algorithm:PackageName.AlgoClassName
```

Configuring Directory Proxy Server for Distribution of Suffix Data

A simple case of data distribution consists of storing entries having UIDs beginning with A through M in one set of directories, and storing entries having UIDs beginning with `N` through `Z` in another set of directories. Directory Proxy Server receives all client operations. The server must determine which set of directories handles A through M, and which set handles N through Z.

The key stages in configuring Directory Proxy Server to handle this data distributions scenario are as follows.

1. Add directories as data sources for Directory Proxy Server.
2. Add the data sources to data source pools to handle the different data distributions.
3. Create data views designed to distribute client requests to the appropriate data pools.
4. Split the LDIF to be loaded into the appropriate data sources.
5. Import the split LDIF into the appropriate data sources.
6. Adjust the operation based weights for the data sources attached to the appropriate data pools.

The following example involves Directory Proxy Server, listening on port 9389. To keep the example simple, the proxy is configured here to distribute as described across only three Directory Server instances. For availability and read scalability, use replicated directory topologies to store LDAP data. One Directory Server instance, `dsA-M:1389` handles the user entries having UIDs beginning with A through M. Another Directory Server instance, `dsN-Z:2389`, handles the user entries having UIDs beginning with N through Z. A final directory instance handles the base entries of the suffix, `dsBase:3389`.

The first step creates and enables the data sources. The base data source holds entries near the root of the suffix that do not have UIDs. In a typical deployment, these entries would be much fewer in number than distributed entries.

```
$ dpconf create-ldap-data-source -p 9389 dsA-M localhost:1389
$ dpconf set-ldap-data-source-prop -p 9389 dsA-M is-enabled:true
```

```
$ dpconf create-ldap-data-source -p 9389 dsN-Z localhost:2389
$ dpconf set-ldap-data-source-prop -p 9389 dsN-Z is-enabled:true

$ dpconf create-ldap-data-source -p 9389 dsBase localhost:3389
$ dpconf set-ldap-data-source-prop -p 9389 dsBase is-enabled:true
```

The second step adds the data sources to a data source pool.

```
$ dpconf create-ldap-data-source-pool -p 9389 "Base Pool"
$ dpconf attach-ldap-data-source -p 9389 "Base Pool" dsBase

$ dpconf create-ldap-data-source-pool -p 9389 "A-M Pool"
$ dpconf attach-ldap-data-source -p 9389 "A-M Pool" dsA-M

$ dpconf create-ldap-data-source-pool -p 9389 "N-Z Pool"
$ dpconf attach-ldap-data-source -p 9389 "N-Z Pool" dsN-Z
```

The third step creates data views designed to distribute client requests to the appropriate data pools. Notice how the base pool handles `dc=example,dc=com`, whereas the pools holding data distributed according to UID values handle `ou=people,dc=example,dc=com`. This step requires a server restart.

```
$ dpconf create-ldap-data-view -p 9389 "Base View" "Base Pool" \
dc=example,dc=com

$ dpconf create-ldap-data-view -p 9389 "A-M View" "A-M Pool" \
ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop -p 9389 "A-M View" \
distribution-algorithm:lexicographic lexicographic-attrs:uid \
lexicographic-lower-bound:a lexicographic-upper-bound:m
The proxy server will need to be restarted in order for the changes to take effect

$ dpconf create-ldap-data-view -p 9389 "N-Z View" "N-Z Pool" \
ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop -p 9389 "N-Z View" \
distribution-algorithm:lexicographic lexicographic-attrs:uid \
lexicographic-lower-bound:n lexicographic-upper-bound:z
The proxy server will need to be restarted in order for the changes to take effect
$ dpadm restart /local/dps
```

The fourth step splits the LDIF to be loaded into the appropriate data sources. This example uses both the `dpadm split-ldif` command to perform the initial split, and also some file editing to retain the top entry in all the data sources. This makes it possible both to retain the top entry that specifies access control instructions, and to use a single import command for each data source.

```
$ dpadm split-ldif /local/dps /opt/SUNWdsee7/resources/ldif/Example.ldif /tmp
```

This step also requires a top entry that is added to the LDIF before import.

```
$ cp /opt/SUNWdsee7/resources/ldif/Example.ldif /tmp/top.ldif
$ vi /tmp/top.ldif
dn: dc=example,dc=com
objectclass: top
objectclass: domain
dc: example
aci: (target ="ldap:///dc=example,dc=com")(targetattr !=
 "userPassword")(version 3.0;acl "Anonymous read-search access";
 allow (read, search, compare)(userdn = "ldap:///anyone");)
aci: (target="ldap:///dc=example,dc=com") (targetattr =
  "*")(version 3.0; acl "allow all Admin group"; allow(all) groupdn =
  "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)

$ cat /tmp/top.ldif /tmp/base\ view.ldif > /tmp/top\ and\ base\ view.ldif
$ cat /tmp/top.ldif /tmp/a-m\ view.ldif > /tmp/top\ and\ a-m\ view.ldif
$ cat /tmp/top.ldif /tmp/n-z\ view.ldif > /tmp/top\ and\ n-z\ view.ldif
```

The fifth step imports the split LDIF into the appropriate data sources. Here, the directory handling the base entries is on port 3389. The directory handling A-M is listening on port 1389. The directory handling N-Z is listening on port 2389.

```
$ dsconf import -p 1389 /tmp/top\ and\ a-m\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).

$ dsconf import -p 2389 /tmp/top\ and\ n-z\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).
$ dsconf import -p 3389 /tmp/top\ and\ base\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).
```

The sixth step adjusts the operation based weights for the data sources attached to the appropriate data pools. If client applications perform operations other than searches, then weights must be set for those operations as well.

```
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Base Pool" dsBase search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "A-M Pool" dsA-M search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "N-Z Pool" dsN-Z search-weight:1
```

After the operations based weights are set, client applications can search through Directory Proxy Server as if the data were not physically distributed.

The following search looks for a user whose UID begins with R.

```
$ ldapsearch -p 9389 -b dc=example,dc=com uid=rfisher
version: 1
dn: uid=rfisher, ou=People, dc=example,dc=com
cn: Randy Fisher
sn: Fisher
givenName: Randy
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Human Resources
ou: People
l: Cupertino
uid: rfisher
mail: rfisher@example.com
telephoneNumber: +1 408 555 1506
facsimileTelephoneNumber: +1 408 555 1992
roomNumber: 1579
```

The next search looks for one of the base entries.

```
$ ldapsearch -p 9389 -b ou=groups,dc=example,dc=com cn=hr\ managers
version: 1
dn: cn=HR Managers,ou=groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Managers
ou: groups
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
uniqueMember: uid=cschmith, ou=People, dc=example,dc=com
description: People who can manage HR entries
```

Creating and Configuring Data Views for Example Use Cases

This section contains the following information about data views and how to create and configure them:

- "Data Views That Provide a Single Point of Access When Different Parts of a Subtree Are Stored in Different Data Sources" on page 427
- "Data Views With Hierarchy and a Distribution Algorithm" on page 428

The examples in this section assume that the connection handler allows all client connections to be processed by Directory Proxy Server.

Data Views That Provide a Single Point of Access When Different Parts of a Subtree Are Stored in Different Data Sources

This section describes how to configure a data view that provides a single point of access to different parts of a subtree. This example contains two data views with the same base DN. A numeric distribution algorithm is used to separate entries into different data views. A data source pool is configured for each set of data-equivalent data sources. The following figure shows the example deployment.

For information about this type of deployment, see "Data Views to Route Requests When Different Parts of a Subtree Are Stored in Different Data Sources" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

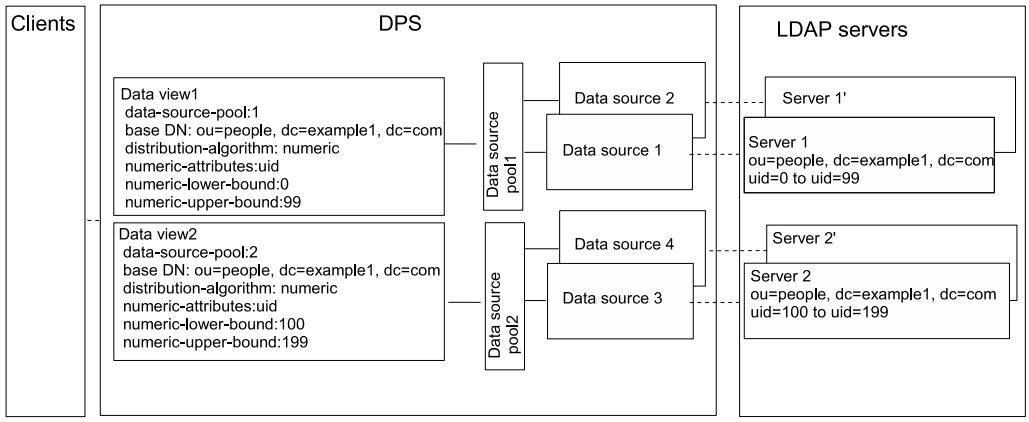


FIGURE 21–1 Sample Deployment That Provides a Single Point of Access When Different Parts of a Subtree Are Stored in Different Data Sources

▼ To Configure Data Views That Provide a Single Point of Access When Different Parts of a Subtree Are Stored in Different Data Sources

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Create a data source for each LDAP server as described in "Creating and Configuring LDAP Data Sources" on page 373.**

2. **Create two data source pools as described in "Creating and Configuring LDAP Data Source Pools" on page 376.**

3. **Attach the data sources that contain one part of the subtree to `data-source-pool-1`, and the data sources that contain the other part of the subtree to `data-source-pool-2`, as described in "Attaching LDAP Data Sources to a Data Source Pool" on page 377.**

4. **(Optional) Configure load balancing.**

   For information, see "Configuring Load Balancing" on page 405.

5. **Create a data view with a distribution algorithm to select entries in `ou=people,dc=example,dc=com` with `uid` between `0` and `99`, and configure the data view to direct requests to `data-source-pool-1`.**

   ```
   $ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
    ldap-data-source-pool:data-source-pool-1 base-dn:ou=people,dc=example,dc=com \
    distribution-algorithm    :numeric numeric-attrs:uid numeric-lower-bound    :0 \
    numeric-upper-bound    :99
   ```

6. **Create another data view with a distribution algorithm to select entries in `ou=people,dc=example,dc=com` with `uid` between `100` and `199`, and configure the data view to direct requests to `data-source-pool-2`.**

   ```
   $ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
    ldap-data-source-pool:data-source-pool-2 base-dn:ou=people,dc=example,dc=com \
    distribution-algorithm:numeric numeric-attrs:uid numeric-lower-bound:100
    numeric-upper-bound    :199
   ```

   The other properties of the data views are the same as the default data view in "Default Data View" on page 387.

7. **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

   For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Data Views With Hierarchy and a Distribution Algorithm

This section describes how to configure a data view to combine hierarchy with distribution algorithms. For information about this type of deployment, see "Data Views With Hierarchy and a Distribution Algorithm" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The example in this section contains four data views. The base DN of data view 1 is superior to the base DNs of the other data views. Data view 3 and data view 4 have the same base DN, but a numeric distribution algorithm separates the entries into different data views.

Directory Proxy Server automatically excludes a subordinate branch of a subtree from a data view when the subordinate branch is configured as the base DN of a separate data view. A numeric distribution algorithm separates entries from the same subtree into different data views. A data source pool is configured for each set of data-equivalent data sources.

The following figure shows the example deployment.

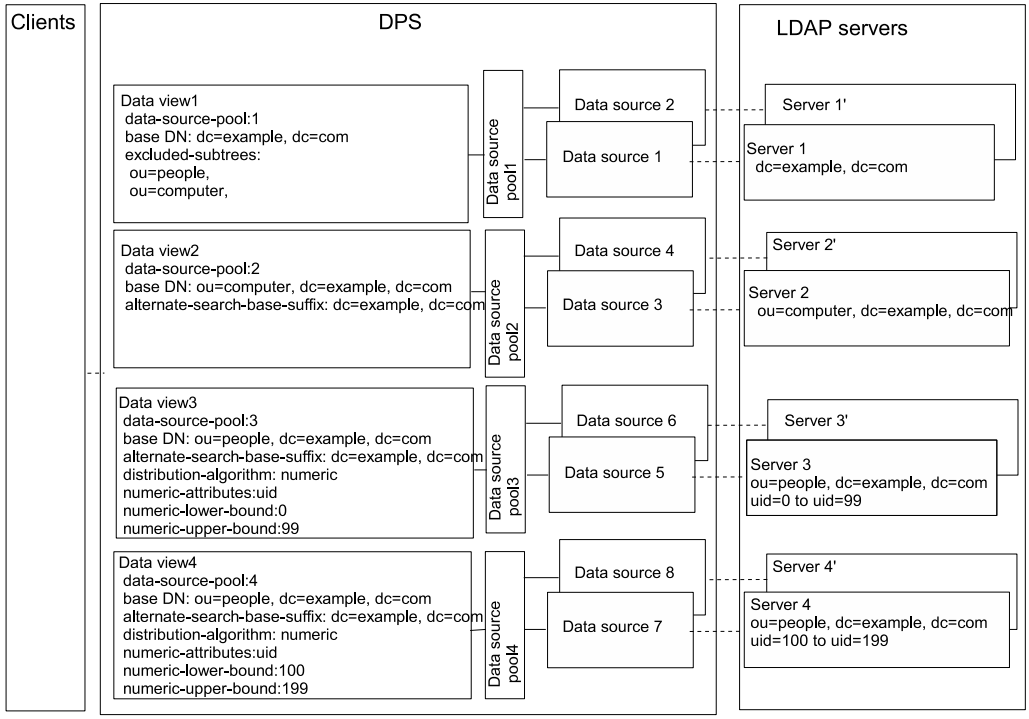


FIGURE 21–2 Sample Data View With Hierarchy and a Distribution Algorithm

▼ To Configure Data Views With Hierarchy and a Distribution Algorithm

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1. **Create a data source for each LDAP server as described in "Creating and Configuring LDAP Data Sources" on page 373.**

2. **Create four data source pools as described in "Creating and Configuring LDAP Data Source Pools" on page 376.**

3. **Attach the data sources to the data source pools by following the instructions in "Attaching LDAP Data Sources to a Data Source Pool" on page 377.**

   - Attach the data sources that contain `dc=example,dc=com` to `data-source-pool-1`.
   - Attach the data sources that contain `ou=computer,dc=example,dc=com` to `data-source-pool-2`.
   - Attach the data sources that contain entries in `ou=people,dc=example,dc=com` with `uid` between `0` and `99` to `data-source-pool-3`.
   - Attach the data sources that contain entries in `ou=people,dc=example,dc=com` with `uid` between `100` and `199` to `data-source-pool-4`.

4. **(Optional) Configure load balancing.**

   For information, see "Configuring Load Balancing" on page 405.

5. **Create a data view with a base DN at `dc=example,dc=com`, that refers to `data-source-pool-1`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
 data-source-pool-1 dc=example,dc=com
```

6. **Create a data view with a base DN at `ou=computer,dc=example,dc=com` that refers to `data-source-pool-2`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
 data-source-pool-2 ou=computer,dc=example,dc=com
```

7. **Create a data view with a base DN at `ou=people,dc=example,dc=com` that refers to `data-source-pool-3`. Configure a distribution algorithm on the data view to select entries with `uid` between `0` and `99`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
 data-source-pool-3 ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop dataview-3 distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:0 numeric-upper-bound:99
```

8. **Create a data view with a base DN at `ou=people,dc=example,dc=com` that refers to `data-source-pool-4`, and configure a distribution algorithm on the data view to select entries with `uid` between `100` and `199`.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-4 \
 data-source-pool-4 ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop dataview-4 distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:100 numeric-upper-bound:199
```

9. **Verify that the subtrees `ou=computer,dc=example, dc=com` and `ou=people,dc=example, dc=com` have been excluded from `dataview-1` by looking at the `excluded-subtrees` parameter.**

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
```

   The list of excluded subtrees is returned.

10 Restart the instance of Directory Proxy Server for the changes to take effect.

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

◆ ◆ ◆ CHAPTER 22

Directory Proxy Server Virtualization

This chapter describes how to create virtual data views. *Virtual data views* transform the source data and present a different view of that data to client applications. Virtual data views include transformed LDAP data views, LDIF data views, join data views, and JDBC™ data views. For an overview of the features of virtual data views and a description of example use cases, see Chapter 18, "Directory Proxy Server Virtualization," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

You cannot use Directory Service Control Center (DSCC) to perform the procedures in this chapter. You must use the command line.

This chapter covers the following topics:

Creating and Configuring LDIF Data Views

An LDIF data view is a simple virtual data view in which an LDIF file is made to look like an LDAP data source. Unlike for LDAP data views, you do not create data sources or data source pools when you set up LDIF data views. Instead, you specify an LDIF file when you create the data view. By default, you cannot write to an LDIF data view. For more information, see "Defining Access Control on Virtual Data Views" on page 435.

For information about creating and configuring LDIF data views, see the following procedures.

▼ To Create an LDIF Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Create an LDIF data view.

```
$ dpconf create-ldif-data-view -h host -p port view-name path-to-ldif-file suffix-dn
```

2 (Optional) View the list of LDIF data views.

```
$ dpconf list-ldif-data-views -h host -p port
```

The `virtual access controls` data view is the only default LDIF data view. This data view is generated by the server and enables requests to be routed to virtual access control instructions (ACIs).

▼ To Configure an LDIF Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 View the properties of an LDIF data view.

```
$ dpconf get-ldif-data-view-prop -h host -p port view-name
```

An LDIF data view has the following default properties:

```
alternate-search-base-dn                  :  -
attr-name-mappings                        :  none
base-dn                                   :  suffixDN
bind-pwd-attr                             :  userPassword
contains-shared-entries                   :  false
custom-distribution-algorithm             :  none
db-pwd-encryption                         :  clear-text
description                               :  -
distribution-algorithm                    :  none
dn-join-rule                              :  none
dn-mapping-attrs                          :  none
dn-mapping-source-base-dn                 :  none
excluded-subtrees                         :  -
filter-join-rule                          :  none
is-enabled                                :  true
is-read-only                              :  false
is-routable                               :  true
ldif-data-source                          :  /path/to/filename.ldif
lexicographic-attrs                       :  all
lexicographic-lower-bound                 :  none
```

```
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  none
non-writable-attr                           :  none
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

2 Change one or more of the properties that are listed in Step 1.

```
$ dpconf set-ldif-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

For example, to change the source LDIF file for the data view, set the `ldif-data-source` property.

```
$ dpconf set-ldif-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" \
myLDIFDataView ldif-data-source:/local/files/example.ldif
```

Defining Access Control on Virtual Data Views

ACIs on virtual data views can be stored in an LDAP directory or in an LDIF file. For information about how virtual ACIs work, see "Access Control On Virtual Data Views" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

When you create a Directory Proxy Server instance, the following default configuration for virtual access controls is defined:

- An LDIF file in which ACIs are stored by default (*instance-path*`/config/access_controls.ldif`)
- An LDIF data view named `virtual access controls`

  This data view enables Directory Proxy Server to access the ACIs stored in the LDIF file.

▼ To Define a New ACI Storage Repository

If you do not want to use the default ACI configuration described previously, you can define a different storage repository.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Create a data view for the repository in which the virtual ACIs will be stored.**

- If the ACIs will be stored in an LDAP directory, create an LDAP data source, an LDAP data source pool, and an LDAP data view, as described in Chapter 18, "LDAP Data Views."
- If the ACIs will be stored in an LDIF file, create an LDIF data view, as described in "Creating and Configuring LDIF Data Views" on page 433.

2 **Specify the name of the data view created in the previous step as the ACI data view.**

```
$ dpconf set-virtual-aci-prop -h host -p port aci-data-view:data-view-name
```

3 **If the ACI repository is an LDAP directory, define the credentials required to access the ACI data view.**

```
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-dn:bind-dn
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-pwd-file:filename
```

▼ To Configure Virtual Access Controls

Regardless of the ACI repository that you use, you must configure the virtual access controls.

Note – Only the Proxy Manager can create a pool of ACIs and manage ACIs directly through the ACI data view. If the ACI repository is an LDAP directory, you must modify the schema of that directory to include the `aciSource` object class and the `dpsaci` attribute. For more information about customizing the schema, see "Extending Directory Server Schema" on page 301.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Create a pool of ACIs in the ACI repository, and set up global ACIs.**

For information about global ACIs, see "Global ACIs" in *Sun Directory Server Enterprise Edition 7.0 Reference*. To set up global ACIs, add an `aciSource` entry under the view base of the ACI data view. For example:

```
% ldapmodify -p port -D "cn=proxy manager" -w -
dn: cn=aci-source-name,cn=virtual access controls
changetype: add
objectclass: aciSource
dpsaci: (targetattr="*") (target="ldap:///ou=people,o=virtual") (version 3.0;
 acl "perm1"; allow(all) groupdn="ldap:///cn=virtualGroup1,o=groups,o=virtual";)
cn: aci-source-name
```

2 **Configure one or more connection handlers to use this pool of ACIs.**

```
% dpconf set-connection-handler-prop -h host -p port connection-handler \
aci-source:aci-source-name
```

3 **Add the required ACIs to the data.**

To do this, create a virtual entry that contains the ACIs. For example:

```
% ldapmodify -p port -D "cn=virtual application,ou=application users,dc=com" -w -
dn: ou=people,o=virtual
changetype: modify
add: dpsaci
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(all) userdn="ldap:///self";)
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(search, read, compare)
 userdn ="ldap:///anyone";)
```

Note – Any user with the appropriate access rights can add and retrieve virtual ACIs through the data view.

Defining Schema Checking on Virtual Data Views

Generally, for LDAP data views, schema checking is performed by the back-end directory, using the back-end directory's schema. Use the following procedure if you want schema checking to be performed by Directory Proxy Server.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

To normalize requests, particularly the DN, set the `use-external-schema` property of the server, as follows:

▼ To Define Schema Checking

1 **Indicate that the server instance should use an external schema.**

```
$ dpconf set-server-prop -h host -p port use-external-schema:true
```

2 **Enable schema checking on the connection handler.**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler \
 schema-check-enabled:true
```

3 **Create a data view that exposes `cn=schema`.**

If the external schema is defined in an LDAP directory, create an LDAP data view, as described in Chapter 18, “LDAP Data Views,” with a view base of `cn=schema`.

If the external schema is defined in an LDIF file, create an LDIF data view, as described in "Creating and Configuring LDIF Data Views" on page 433 with a view base of `cn=schema`.

4 **Add this data view to the list of data views exposed by the connection handler.**

By default, all data views are exposed by the connection handler. If you have defined a custom list of data views that are exposed by the connection handler, add this data view to the list.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler \
 data-view-routing-custom-list+:data-view-name
```

Creating and Configuring Join Data Views

A join data view is an aggregation of multiple data views. For information about how a join data view works, see "Join Data Views" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

For information about how to create and configure join data views, see the following procedures.

▼ To Create a Join Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Identify the primary and secondary data views that will be aggregated to form the join view.**

The primary and secondary data views must exist before the join view can be created. The primary and secondary views can be any type of data view, including an LDAP data view, LDIF data view, JDBC data view, or another join data view. Specific properties must be configured on the secondary view to allow it to function as the source for a join view. For more information, see "To Configure the Secondary View of a Join View" on page 441.

2 **Create the join data view.**

```
$ dpconf create-join-data-view -h host -p port view-name primary-view secondary-view \
 suffix-dn
```

3 **(Optional) View the list of join views to check that your data view has been created successfully.**

```
$ dpconf list-join-data-views -h host -p port
```

▼ To Configure a Join Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 View the properties of a join data view.

```
$ dpconf get-join-data-view-prop -h host -p port view-name
```

The default properties of a join data view are as follows:

```
allow-heuristic-search                      :  true
allow-partial-search                        :  false
alternate-search-base-dn                    :  -
attr-name-mappings                          :  none
base-dn                                     :  suffixDN
contains-shared-entries                     :  false
custom-distribution-algorithm               :  none
description                                 :  -
distribution-algorithm                      :  none
dn-join-rule                                :  none
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  none
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
join-rule-control-enabled                   :  false
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  none
non-writable-attr                           :  none
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-dn-regular-expression :  all
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
primary-view                                :  primary-view
process-bind                                :  -
replication-role                            :  master
request-grouping-size                       :  5
secondary-view                              :  secondary-view
viewable-attr                               :  all except non-viewable-attr
vlv-control-enabled                         :  false
vlv-control-page-size                       :  1k
vlv-control-sorting-attr                    :  objectclass
writable-attr                               :  all except non-writable-attr
```

2 **Change one or more of the properties that are listed in Step 1.**

```
$ dpconf set-join-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

For example, to change the primary data view of a data source to `myLDAPDataView`, use the following command:

```
$ dpconf set-join-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" \
 myJoinDataView primary-view:myLDAPDataView
```

If `vlv-control-enabled` is set to `true`, Directory Proxy Server uses VLV control in search requests when it contacts the primary data view.

3 **When a join data view is configured, set `viewable-attr` and `writable-attr` properties on primary data view and secondary data view.**

Setting of these properties helps in splitting the search filters appropriately on primary and secondary data views. Otherwise, there might be discrepancies in search results when search filter contains attributes from secondary data view.

4 **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

▼ To Configure a Join Data View to Enable Referencing of a Data View by Multiple Join Data Views

Setting join rule configuration information in the join data view makes the data view to be referenced by multiple join data views. To do so, perform the following:

1 **Set `join-rule-control-enabled` to `true` on the join data view.**

```
$ dpconf set-join-data-view-prop view-name join-rule-control-enabled:true
```

After setting `join-rule-control-enabled` to `true`, join rule configuration information stored in the join data view is used by the server. If you have a join data view with the join rule configuration information stored in the secondary data view then this information is not used by the server. To have this information used by the server, you will have to manually add the configuration information at the join data view level.

2 **Define a join rule that determines how the secondary view is related to the primary view.**

The join rule can be one of the following:

- DN join rule

```
$ dpconf set-join-data-view-prop view-name \
dn-join-rule:uid=\${primary-view-name.uid},ou=People,dc=example
```

- Filter join rule

```
$ dpconf set-join-data-view-prop view-name \
filter-join-rule:uid=\${primary-view-name.uid}
```

In the above commands, the attribute name is enclosed in `${}` when treated as a variable. If you do not use attribute names enclosed in `${}`, the attribute names are treated as constants.

If you use bash or ksh in UNIX, the `$` character should be escaped by `\` in the `\${`*primary-view-name*`.uid}` like constructions whereas no escaping is required on Windows.

▼ To Configure the Secondary View of a Join View

Specific properties must be configured on the secondary data view to allow it to function as the source for a join view. Because the secondary view can be any type of data view, the command you use will depend on the data view type. The following sample commands assume that the secondary view is an LDAP data view. For more information about the properties described here, see "Additional Secondary Data View Properties" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Define a join rule that determines how the secondary view is related to the primary view.

Never set the `filter-join-rule` and `dn-join-rule` on the primary data view of a join view.

The join rule can be one of the following:

- DN join rule

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
dn-join-rule:uid=\${primary-view-name.uid},ou=People,dc=example
```

- Filter join rule

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
filter-join-rule:uid=\${primary-view-name.uid}
```

The configuration for the `dn-join-rule` and `filter-join-rule` properties is used by the server only if the `join-rule-control-enabled` property on the join data view is set to `false`. Otherwise, if the `join-rule-control-enabled` property is set to `true` on the join data view, then the information set on the secondary view will be ignored.

2 If the filter join rule is set on the join data view, you need to set a virtual transformation rule on the secondary data view to be able to add an entry on the join data view.

```
dpconf add-virtual-transformation secondary-view-name \
write add-attr-value dn uid=\${uid}
```

Note – Without setting this rule, addition of entries to join data view would not be possible.

3 **(Optional) Specify whether binds are allowed on the secondary view.**

By default, binds are permitted on all data views. If you want to prohibit binds to the secondary data view, run the following command:

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name process-bind:false
```

For more information about this property, see "Handling of Binds" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

4 **(Optional) Specify whether the secondary view contains shared entries.**

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
contains-shared-entries:true
```

For more information about this property, see "Handling of Shared Entries" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

Creating and Configuring Coordinator Data Views

The Coordinator data view has an ordered list of data views to coordinate. When the Coordinator data view receives a request, it sends the request to its coordinated data views in the right order, which is defined by the order used in the CLI to specify the coordinated data view. The Coordinator data view sends the request to coordinated data view as defined by the routing policy and in the end returns the result(s) to the client. For more information, see "Coordinator Data Views" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The Coordinator data view supports the following routing policies:

- `first-match`: stops sending the request at first match
- `all-candidates`: sends the request to all the coordinated data views
- `mirror`: sends all the requests to first coordinated data view but the write requests are sent to all the coordinated data views.

Note – All the data views coordinated by a coordinator data view must have the same view base identical with the view base of the coordinator data view.

▼ To Create a Coordinator Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Identify the data views that will be aggregated to form the Coordinator data view.**

The data views must exist before the Coordinator data view can be created. The data views can be any type of data view, including an LDAP data view, LDIF data view, JDBC data view, or another join data view.

2 **Create the Coordinator data view.**

```
$ dpconf create-coordinator-data-view -h host -p port VIEW_NAME \
COORDINATED_VIEW [COORDINATED_VIEW...] SUFFIX_DN
```

The following command creates a Coordinator data view, `view_com`, using two data views.

```
$ dpconf create-coordinator-data-view -h host -p port view_com \
first_view second_view dc=com
```

3 **(Optional) View the list of Coordinator data views to check that your data view has been created successfully.**

```
$ dpconf list-coordinator-data-views -h host -p port
```

In this case, the above command displays `view_com`.

▼ To Configure a Coordinator Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **View the properties of a Coordinator data view.**

```
$ dpconf get-coordinator-data-view-prop -h host -p port VIEW_NAME
```

The default properties of a join data view are as follows:

```
alternate-search-base-dn                    :  ""
attr-name-mappings                          :  none
base-dn                                     :  ""
contains-shared-entries                     :  -
coordinated-data-view                       :  first_view
coordinated-data-view                       :  second_view
custom-distribution-algorithm               :  none
description                                 :  -
distribution-algorithm                      :  none
dn-join-rule                                :  none
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  ""
filter-join-rule                            :  none
is-enabled                                  :  true
is-read-only                                :  false
```

```
is-routable                                 :  true
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  none
non-writable-attr                           :  none
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
routing-policy                              :  all-candidates
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

For example, the following commands lists the `coordinated-data-views` property of the specified Coordinator data view.

```
$ dpconf get-coordinator-data-view-prop VIEW_NAME coordinated-data-view

coordinated-data-view  :  first_view
coordinated-data-view  :  second_view
```

2 Change one or more of the properties that are listed in Step 1.

```
$ dpconf set-coordinator-data-view-prop -h host -p port VIEW_NAME PROP:VAL \
[PROP:VAL...]
```

For example, add more coordinated data views to the Coordinator data view using the following command:

```
$ dpconf set-coordinator-data-view-prop -h host -p port view_com \
coordinated-data-view+:third_view coordinated-data-view+:fouth_view
```

As per the order of the coordinated data views specified in the above command, the coordinated data views are send in the following order:

```
$ dpconf get-coordinator-data-view-prop VIEW_NAME coordinated-data-view

coordinated-data-view  :  first_view
coordinated-data-view  :  second_view
coordinated-data-view  :  third_view
coordinated-data-view  :  fourth_view
```

3 **(Optional) Change the `routing-policy` mode that describe how Coordinator data view sends the requests to coordinated data views.**

```
$ dpconf set-coordinator-data-view-prop -h host -p port view_com \
routing-policy:first-match
```

For more information, see `routing-policy(5dpconf)`.

4 **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Creating and Configuring JDBC Data Views

A JDBC data view enables you to make a relational database accessible to LDAP client applications. For information about how JDBC data views work, see "JDBC Data Views" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

For information about how to create and configure JDBC data views, see the following procedures.

▼ To Create a JDBC Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Create a JDBC data source for the relational database.**

```
$ dpconf create-jdbc-data-source -h host -p port -b db-name -B db-url -J driver-url \
[-J driver-url]... -S driver-class  source-name
```

Currently, only one JDBC data source is supported for each JDBC data view. In other words, you cannot load balance across JDBC data sources. To access multiple JDBC data sources, you can create a data view for each data source, and join them together with a join data view.

The following properties must be set when you create a JDBC data source:

`db-name` The name of the relational database, for example, `payrolldb`.

`db-url` The URL to the database, in the form `jdbc:`*vendor*`:`*driver*`://`*dbhost*`:`*dbport*.

The `db-url` is *not* a complete JDBC database URL, because it does not contain the database name. (The database name is specified by the `db-name` property.)

You must finish `db-url` with a `/` for MySQL, DB2, and Derby databases and with a `:` for Oracle database.

`driver-class` The JDBC driver class, for example `org.hsqldb.jdbcDriver`.

`driver-url` The path to the JDBC driver, for example `file:///path/to/hsqldb/lib/hsqldb.jar`.

The `driver-url` property is multivalued. Hence, `driver-url` can support multiple JAR files for the JDBC driver to ensure connectivity to the JDBC source on different platforms.

If a 3rd party JDBC driver, which is other than the JDBC driver provided by the DB Vendor, is used to connect to the RDBMS back-end, set the `db-vendor` property. For more information about the `db-vendor` property, see `db-vendor(5dpconf)`

2 Create a JDBC data source pool.

```
$ dpconf create-jdbc-data-source-pool -h host -p port pool-name
```

3 Attach the JDBC data source to the JDBC data source pool.

```
$ dpconf attach-jdbc-data-source -h host -p port pool-name source-name
```

4 Create a JDBC data view.

```
$ dpconf create-jdbc-data-view -h host -p port view-name pool-name suffix-DN
```

5 (Optional) View the list of JDBC data views to check that your data view has been created successfully.

```
$ dpconf list-jdbc-data-views -h host -p port
```

▼ To Configure a JDBC Data View

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 View the properties of a JDBC data view.

```
$ dpconf get-jdbc-data-view-prop -h host -p port view-name
```

The default properties of a JDBC data view are as follows:

```
alternate-search-base-dn                    :  -
attr-name-mappings                          :  none
base-dn                                     :  o=sql1
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
```

```
dn-mapping-source-base-dn                  :  none
excluded-subtrees                          :  -
filter-join-rule                           :  -
is-enabled                                 :  true
is-read-only                               :  false
is-routable                                :  true
jdbc-attr-date-format                      :  yyyy-MM-dd
jdbc-attr-time-format                      :  hh:mm:ss
jdbc-attr-timestamp-format                 :  yyyy-MM-dd hh:mm:ss
jdbc-data-source-pool                      :  pool-name
lexicographic-attrs                        :  all
lexicographic-lower-bound                  :  none
lexicographic-upper-bound                  :  none
non-viewable-attr                          :  -
non-writable-attr                          :  -
numeric-attrs                              :  all
numeric-default-data-view                  :  false
numeric-lower-bound                        :  none
numeric-upper-bound                        :  none
pattern-matching-base-object-search-filter :  all
pattern-matching-dn-regular-expression     :  all
pattern-matching-one-level-search-filter   :  all
pattern-matching-subtree-search-filter     :  all
process-bind                               :  -
replication-role                           :  master
viewable-attr                              :  all except non-viewable-attr
writable-attr                              :  all except non-writable-attr
```

2 Change one or more of the properties that are listed in Step 1.

```
$ dpconf set-jdbc-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

3 (Optional) In addition to the ISO format, you can set JDBC attributes of type date, time, and timestamp attributes in all of the following formats as well.

Following lists the components that constitutes date and time:

Letter	Date or Time Component	Examples
G	Era designator	AD
y	Year	1996; 96
M	Month in year	July; Jul; 07
w	Week in year	27
W	Week in month	2
D	Day in year	189
d	Day in month	10
F	Day of week in month	2
E	Day in week	Tuesday; Tue
a	Am/pm marker	PM

```
H           Hour in day (0-23)       0
k           Hour in day (1-24)       24
K           Hour in am/pm (0-11)     0
h           Hour in am/pm (1-12)     12
m           Minute in hour           30
s           Second in minute         55
S           Millisecond              978
z           Time zone                Pacific Standard Time; PST; GMT-08:00
Z           Time zone                -0800
```

The following examples show how date and time patterns are interpreted in the U.S. locale. The given date and time are `2001-07-04 12:08:56` local time in the U.S. Pacific Time time zone.

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2001.07.04 AD at 12:08:56 PDT
"EEE, MMM d, ''yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o''clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02001.July.04 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 4 Jul 2001 12:08:56 -0700
"yyMMddHHmmssZ"	010704120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2001-07-04T12:08:56.235-0700

▼ To Configure JDBC Tables, Attributes, and Object Classes

When you configure a JDBC data view, you must also configure the following objects:

- **JDBC object class.** Maps one or more JDBC tables to an LDAP object class.
- **JDBC table.** Defined for each relational database table.
- **JDBC attribute.** Defines an LDAP attribute from a specified column in a JDBC table.

1 Create a JDBC table for each table in the relational database.

```
% dpconf create-jdbc-table jdbc-table-name db-table
```

The name of the *db-table* is case sensitive. Make sure that you use the identical case that is used in the relational database, otherwise operations that target that table might fail.

2 Create a JDBC attribute for each column in each relational database table.

```
% dpconf add-jdbc-attr table-name attr-name sql-column
```

Creating a JDBC attribute maps the table column to an LDAP attribute.

3 (Optional) If the column in the relational database is case sensitive, change the LDAP syntax of the JDBC attribute.

```
% dpconf set-jdbc-attr-prop table-name attr-name ldap-syntax:ces
```

The value of `ldap-syntax` is `cis` by default. This implies that the `jdbc-attr` is case insensitive. Change the value to `ces` if your relational database is case sensitive.

Certain relational databases, such as Oracle and DB2, are case sensitive by default. LDAP is case insensitive by default. When Directory Proxy Server detects that a column of the relational database table is case sensitive, an `ldapsearch` query with the corresponding attribute in the filter is translated into a SQL query using the function `UPPER`.

For example, the query `ldapsearch -b "dc=mysuffix" "(attr=abc)"` is translated into the following SQL query:

```
SELECT * FROM mytable WHERE (UPPER(attr)='ABC')
```

By default, this type of query is not indexed. Queries of this nature can therefore have a substantial performance impact.

You can alleviate the performance impact in two ways:

- By setting the `ldap-syntax` property of the jdbc-attr to `ces`.
- By creating an index with the function `UPPER` for each `jdbc-attr` that might be used in an LDAP filter.

Note – If your relational database is not case sensitive, use `ldap-syntax` with the default value, that is, `cis`. `ldap-syntax:ces` is not supported with the case insensitive databases.

---

4 Create a JDBC object class for the LDAP relational database table.

```
% dpconf create-jdbc-object-class view-name objectclass primary-table \
  [secondary-table... ] DN-pattern
```

Creating a JDBC object class essentially specifies an LDAP object class with which these tables will be associated. The JDBC object class also specifies the primary table and the secondary tables, if they exist.

When you create a JDBC object class, you specify a DN pattern. DN pattern describes which attributes are to be used to construct DN of the entry. For example, when you specify DN pattern as `uid` then the DN of the entry is constructed using attribute `uid` and view base of the data view. For example, `uid=bjensen,ou=people,dc=example,dc=com`. The DN pattern can constitute multiple attributes. In that case, attributes should be separated by `,` (comma). For example, if DN pattern is specified as `uid,country`, DN of the entry returned by the data view is `uid=bjensen,country=America,ou=people,dc=example,dc=com`.

When data source contains duplicate entries and the duplicate entries are not required, set `perform-distinct-select` to `true` using the following command:

```
% dpconf set-jdbc-object-class-prop view-name objectclass perform-distinct-select:true
```

All the subtree components defined in the DN pattern of JDBC object class should have a JDBC object class defined for them. For example, if there is a DN pattern `uid,ou` in a JDBC object class, there should be a JDBC object class definition with a DN pattern `ou`. This is necessary for Directory Proxy Server to construct a properly structured DIT. Otherwise, the subtree with values like `ou=xxx,`*base-DN* would not be returned in the search results.

5 If a secondary table exists, define the join rule between the primary table and the secondary table.

```
% dpconf set-jdbc-table-prop secondary-table-name filter-join-rule:join-rule
```

A join rule is defined on the secondary table and determines how data from that table is linked to data from the primary table. How you define the relationships between the primary and secondary tables of an object class is important. For more information, see "Defining Relationships Between JDBC Tables" on page 450.

6 Specify the super class for the JDBC object class.

```
% dpconf set-jdbc-object-class-prop view-name objectclass super-class:value
```

The super class indicates the LDAP object class from which the JDBC object class inherits.

Defining Relationships Between JDBC Tables

In the simplest case, a JDBC object class contains only a single (primary) table. There is no secondary table, and thus no need to define relationships between tables.

If the object class contains more than one table, the relationships between these tables must be clearly defined. The relationships between tables are always defined on the secondary table. The following properties of a secondary table enable you to define these relationships:

- `is-single-row-table` specifies that an LDAP entry has only one matching row in the table.
- `contains-shared-entries` specifies that a row in the secondary table is used by more than one row in the primary table.
- `filter-join-rule` indicates how an entry should be retrieved from the secondary table based on something in the primary table.

The following examples illustrate how the filter join rule is defined, based on the values of the first two properties. These examples assume that the object class has one primary table and one secondary table.

EXAMPLE 22–1 `is-single-row-table:true` and `contains-shared-entries:true`

These are the default values of these properties. In this case, the relationship between the primary and secondary tables is *n->1*, that is, *n* rows in the primary table reference one shared row in the secondary table.

In the relational database, a foreign key (FK) is defined in the primary table, and points to a column in the secondary table.

Take, for example, an organization in which several employees can share the same manager. Two relational database tables are defined, with the following structure:

```
primary table : EMPLOYEE [ID, NAME, FK_MANAGER_ID]
secondary table : MANAGER  [ID, NAME]
```

The following object class and attributes are defined:

```
object-class : employee
attr : name (from primary EMPLOYEE.NAME)
attr : manager (from secondary MANAGER.NAME)
```

The following filter join rule is defined in the secondary table:

```
ID=\${EMPLOYEE.FK_MANAGER_ID}"
```

In case of multiple secondary tables, you must configure `filter-join-rule` on each secondary table. For more information on how to configure `filter-join-rule` for multiple secondary tables, see the Step 12.

With this configuration, the following behavior occurs for LDAP operations:

- **Adding an employee entry.** If the manager in the employee entry does not exist in the table, a new row is created. If the manager does exist, an existing row is used.
- **Replacing the value of the "manager" attribute in an entry.** The value of the row `MANAGER.NAME` is changed.
- **Deleting an employee entry.** The row in the secondary table is not deleted because the manager entries are shared.
- **Deleting the "manager" attribute from an entry.** The row in the secondary table is deleted and the foreign key (`EMPLOYEE.FK_MANAGER_ID`) is set to `NULL`.

  If an attribute is deleted, the value of the attribute is set to NULL. But if the attribute is defined as a NOT NULL attribute in the table definition, the deletion is not possible.

EXAMPLE 22–2 `is-single-row-table:true` and `contains-shared-entries:false`

In this case, the relationship between the primary and secondary tables is *1->1* or *1<-1*, that is, one row in the primary table is referenced by one row in the secondary table.

In the relational database, the foreign key (FK) might be defined in the primary table, or in the secondary table.

Take, for example, an organization in which the UID of employees is stored in one table, and the surname of employees is stored in a second table. Two relational database tables are defined, with the following structure:

```
primary table : UID [ID, VALUE, FK_SN_ID]
secondary table : SN [ID, VALUE]
```

The following object class and attributes are defined:

```
object-class : employee
attr : uid (from primary UID.VALUE)
attr : sn (from secondary ID.VALUE)
```

The following filter join rule is defined in the secondary table:

```
ID=\${UID.FK_SN_ID}
```

You can use it in the following manner using the `dpconf` command:

```
dpconf set-jdbc-table-prop SN filter-join-rule:ID=\${UID.FK_SN_ID}
```

This configuration could be the other way around, with the foreign key `FK_UID_ID` stored in the secondary table, and pointing to `UID.ID`.

EXAMPLE 22–3 `is-single-row-table:false` and `contains-shared-entries:false`

In this case, the relationship between the primary and secondary tables is *1->n*, that is, one row in the primary table is referenced by *n* rows in the secondary table. This example illustrates the case of multi-valued attributes. A multi-valued attribute is represented as a set of rows in the secondary table, with one row per attribute value.

In the relational database, the foreign key is defined in the secondary table, and points to a column in the primary table.

Take, for example, an organization in which an employee can have several telephone numbers. Two relational database tables are defined, with the following structure:

EXAMPLE 22–3 `is-single-row-table:false` and `contains-shared-entries:false` *(Continued)*

```
primary table : EMPLOYEE [ID, NAME]
secondary table : PHONE [ID, VALUE, USER_ID]
```

The following object class and attributes are defined:

```
object-class : employee
attr : cn (from primary EMPLOYEE.NAME)
attr : telephoneNumber (from secondary PHONE.VALUE)
```

The following filter join rule is defined in the secondary table:

```
USER_ID=\${EMPLOYEE.ID}
```

EXAMPLE 22–4 `is-single-row-table:false` and `contains-shared-entries:true`

This case is currently unsupported in Directory Proxy Server.

Sample Virtual Configurations

The following section provides two sample configurations. These configurations illustrate the main features of a virtual directory, and indicate how these features are configured.

Joining an LDAP Directory and a MySQL Database

The procedures in this section describe a sample virtual configuration that joins an LDAP directory and a MySQL database. The LDAP directory is the primary data source, that contains most of the user information. The MySQL database contains additional information about the users. The resulting configuration is illustrated in the following figure.

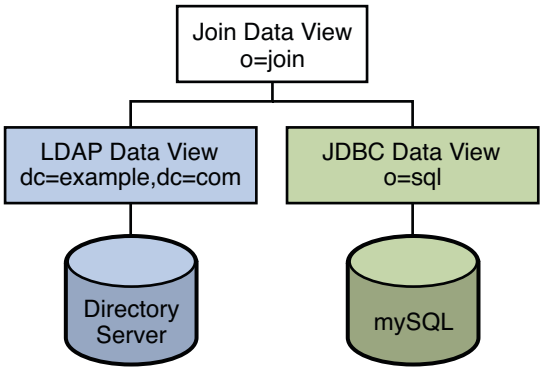


FIGURE 22–1 Sample Virtual Configuration

You can use the sample data provided in *install-path*`/resources/ldif/Example.ldif` to duplicate this example, or you can substitute the sample data with your own data.

This configuration can be broken into three sections:

- Configuring and testing the LDAP data view
- Configuring and testing the JDBC data view
- Configuring and testing the join data view

For simplicity, all the commands in this section assume that the Directory Proxy Server is running on the local host in `/local/dps`. The commands also assume that the following environment variables have been set:

`DIR_PROXY_PORT`	`1389`
`LDAP_ADMIN_PWF`	`pwd.txt`, a file containing the administrator password.
`DIRSERV_PORT`	`4389`
`LDAP_ADMIN_USER`	`cn=Directory Manager`

Configuring and Testing the LDAP Data View

▼ To Configure the LDAP Data View

Before You Begin The tasks in this section assume the following information:

- A Directory Server instance is running on `host1`, on port `4389`.
- Data in the Directory Server is stored under the suffix `dc=example,dc=com`. To duplicate this example, create a Directory Server instance, create the suffix `dc=example,dc=com`, and import the sample data in *install-path*`/resources/ldif/Example.ldif`.

1. **Create an LDAP data source named `myds1` for the Directory Server instance.**

   ```
   % dpconf create-ldap-data-source myds1 host1:4389
   ```

2. **Enable the data source, and allow write operations to the data source.**

   ```
   % dpconf set-ldap-data-source-prop myds1 is-enabled:true is-read-only:false
   ```

3. **Create an LDAP data source pool named `myds1-pool`.**

   ```
   % dpconf create-ldap-data-source-pool myds1-pool
   ```

4. **Attach the LDAP data source to the LDAP data source pool.**

   ```
   % dpconf attach-ldap-data-source myds1-pool myds1
   ```

5. **Specify that the data source should receive 100% of the bind, add, search, and modify operations from that data source pool.**

   ```
   % dpconf set-attached-ldap-data-source-prop myds1-pool myds1 add-weight:100 \
    bind-weight:100 modify-weight:100 search-weight:100
   ```

6. **Create an LDAP data view for the data source pool, named `myds1–view`, with a base DN of `dc=example,dc=com`.**

   ```
   % dpconf create-ldap-data-view myds1-view myds1-pool dc=example,dc=com
   ```

▼ To Test the LDAP Data View

1. **As a user under `dc=example,dc=com`, search all entries in the LDAP data source to verify that you can read from the data view.**

   ```
   % ldapsearch -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery \
    -b dc=example,dc=com "objectclass=*"
   ```

   Note – You must use the credentials of a user under `dc=example,dc=com`. If you want to use `cn=Directory Manager`, you must define a data view to handle that DN.

2. **As a user under `dc=example,dc=com`, modify the `userPassword` attribute to verify that you can write to the data view.**

   ```
   % ldapmodify -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery
   dn: uid=kvaughan,ou=people,dc=example,dc=com
   changetype: modify
   replace: userPassword
   userPassword: myNewPassword
   ```

   Note – A default ACI in Directory Server allows users to modify their own passwords.

Configuring and Testing the JDBC Data View

The following tasks assume that a MySQL database is installed, running and populated with data, and that the MySQL database has the following characteristics:

- Database name : `sample_sql`
- Database URL : `host2.example.com:3306/`
- JDBC driver URL : `file:/net/host2.example/local/mysql/lib/jdbc.jar`
- Driver class : `com.mysql.jdbc.Driver`
- Database user : `root`
- Database password file : `mysqlpwd.txt`

The following table describes the tables in the database, and their composite fields. You need this information to set up the JDBC data view.

MySQL Table	Fields
EMPLOYEE	ID, SURNAME,PASSWORD, ROOM, COUNTRY_ID
COUNTRY	ID, NAME
PHONE	USER_ID, NUMBER

▼ To Configure the JDBC Data View

1 Create a JDBC data source named mysql1 for the SQL database.

```
% dpconf create-jdbc-data-source -b sample_sql \
 -B jdbc:mysql://host2.example.com:3306/ \
 -J file:/net/host2.example/local/mysql/lib/jdbc.jar \
 -S com.mysql.jdbc.Driver mysql1
```

Note – You can set the number of connections for JDBC data source using the `num-connection-incr(5dpconf)`, `num-connection-init(5dpconf)`, and `num-connection-limit(5dpconf)` JDBC data source properties.

2 Specify the user name and password file for the SQL database.

```
% dpconf set-jdbc-data-source-prop mysql1 db-pwd:sqlpwd db-user:rootUser
```

The *sqlpwd* and *rootUser* are the password and username of the authenticating user and these credentials are stored in the database. You must configure the proxy server with username and password when configuring a JDBC data source.

The `db-vendor` property of the JDBC data source should be set using `set-jdbc-data-source-prop` if a third party JDBC driver, which is other than the one

provided by DB Vendor, is used to connect to the RDBMS back-end. This data is used to construct vendor specific SQL statements, whenever possible, to improve performance. For more information, see `db-vendor(5dpconf)`.

3 **Restart the proxy server.**

```
% dpadm restart /local/dps
```

4 **Enable the data source, and allow write operations to the data source.**

```
% dpconf set-jdbc-data-source-prop mysql1 is-enabled:true is-read-only:false
```

5 **(Optional) Set `monitoring-inactivity-timeout`, `monitoring-interval`, and `monitoring-mode` for better monitoring of the open connections and data sources. .**

For more information, see `monitoring-inactivity-timeout(5dpconf)`, `monitoring-interval(5dpconf)`, and `monitoring-mode(5dpconf)`

6 **Create a JDBC data source pool named `mysql1–pool`.**

```
% dpconf create-jdbc-data-source-pool mysql1-pool
```

7 **Attach the JDBC data source to the data source pool.**

```
% dpconf attach-jdbc-data-source mysql1-pool mysql1
```

8 **Create a JDBC data view for the data source pool, named `myjdbc1–view`, with a base DN of `o=sql`.**

```
% dpconf create-jdbc-data-view mysql1-view mysql1-pool o=sql
```

9 **Create a JDBC table for each table in the MySQL database.**

```
% dpconf create-jdbc-table employee1 EMPLOYEE
% dpconf create-jdbc-table country1 COUNTRY
% dpconf create-jdbc-table phone1 PHONE
```

The name of the table in the SQL database is case sensitive. Make sure that you use the same case that is used in the SQL database.

10 **Create a JDBC attribute for each column in each table.**

Creating a JDBC attribute maps the MySQL column to an LDAP attribute.

```
% dpconf add-jdbc-attr employee1 uid ID
% dpconf add-jdbc-attr employee1 sn SURNAME
% dpconf add-jdbc-attr employee1 userPassword PASSWORD
% dpconf add-jdbc-attr employee1 roomNumber ROOM
% dpconf add-jdbc-attr phone1 telephoneNumber NUMBER
% dpconf add-jdbc-attr country1 countryName NAME
```

It is not necessary to create JDBC attributes for the `phone1` `user_id` and `country1` `id` columns, because these columns only contain the values that are present in `EMPLOYEE.ID` for which an LDAP attribute `uid` is already created.

11 Create a JDBC object class for the LDAP `person` object class.

In this step, the `employee1` table is identified as the primary table, and the `country1` and `phone1` tables are identified as secondary tables. The creation of a JDBC object class also requires a DN. In this example, the DN is constructed from the `uid` attribute and the base DN of the data view.

```
% dpconf create-jdbc-object-class mysql1-view person employee1 country1 phone1 uid
```

12 Define the join rules between the primary table and the secondary tables.

A join rule is defined on the secondary table and determines how data from that table is linked to data from the primary table.

```
% dpconf set-jdbc-table-prop country1 filter-join-rule:ID=\${EMPLOYEE.COUNTRY_ID}
% dpconf set-jdbc-table-prop phone1 filter-join-rule:USER_ID=\${EMPLOYEE.ID}
```

13 Specify the super class for the JDBC object class.

The super class indicates the LDAP object class from which the JDBC object class inherits attributes.

```
% dpconf set-jdbc-object-class-prop mysql1-view person super-class:top
```

▼ To Create the Required ACIs

Before you can test the JDBC data view, you must enable write access to the data view by configuring ACIs. By default, write access to non-LDAP data views is denied. For the purposes of this example, it is sufficient to add one global ACI that allows users to modify their own passwords.

1 As the Proxy Manager, add a pool of ACIs to the JDBC data source and add a global ACI that allows users to modify their own entries.

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=mysql1,cn=virtual access controls
changetype: add
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=sql")
(version 3.0; acl "enable all access for all users "; allow(all)
userdn="ldap:///uid=kvaughan,o=sql";)
cn: mysql1
```

2 Create a connection handler to handle connections to the `o=sql` domain.

```
% dpconf create-connection-handler mysql1-handler
```

3 **Enable the connection handler and configure it to handle all binds from users in the `o=sql` domain.**

```
% dpconf set-connection-handler-prop mysql1-handler is-enabled:true \
 bind-dn-filters:"uid=.*,o=sql"
```

4 **Configure the connection handler to use the pool of ACIs added previously.**

```
% dpconf set-connection-handler-prop mysql1-handler aci-source:mysql1
```

▼ To Test the JDBC Data View

1 **As a user under `o=sql`, search the JDBC data source to verify that you can read from the data view.**

```
% ldapsearch -p 1389 -D "uid=kvaughan,o=sql" -w mypwd -b o=sql "objectclass=*"
```

Note – You must use the credentials of a user under `o=sql`.

---

2 **As a user under `o=sql`, modify the `userPassword` attribute to verify that you can write to the data view.**

```
% ldapmodify -p 1389 -D "uid=kvaughan,o=sql" -w mypwd
dn: uid=kvaughan,o=sql
changetype: modify
replace: userPassword
userPassword: myNewpwd
```

Creating and Testing the Join Data View

▼ To Create the Join Data View

1 **Create a join data view named `myjoin1–view`.**

Specifying the LDAP data view as the primary data view, and the JDBC data view as the secondary data view.

```
% dpconf create-join-data-view myjoin1-view myds1-view mysql1-view o=join
```

2 **Define a join rule on the secondary data view.**

The following join rule specifies that the `uid` attribute of entries from the secondary data view should match the `uid` attribute of entries from the primary data view.

```
% dpconf set-jdbc-data-view-prop mysql1-view filter-join-rule:uid=\${myds1-view.uid}
```

3 If the filter join rule is set on the join data view, you need to set a virtual transformation rule on the secondary data view to be able to add an entry on the join data view.

```
dpconf add-virtual-transformation secondary-view-name \
write add-attr-value dn uid=\${uid}
```

Note – Without setting this rule, addition of entries to join data view would not be possible.

4 Define the set of attributes that can be read from and written to the primary data view through a join data view.

```
% dpconf set-ldap-data-view-prop myds1-view viewable-attr:dn \
viewable-attr:cn viewable-attr:sn viewable-attr:givenName \
viewable-attr:objectClass viewable-attr:ou viewable-attr:l \
viewable-attr:uid viewable-attr:mail viewable-attr:telephoneNumber \
viewable-attr:facsimileTelephoneNumber viewable-attr:roomNumber \
viewable-attr:userPassword
% dpconf set-ldap-data-view-prop myds1-view writable-attr:dn \
writable-attr:cn writable-attr:sn writable-attr:givenName \
writable-attr:objectClass writable-attr:ou writable-attr:l \
writable-attr:uid writable-attr:mail writable-attr:telephoneNumber \
 writable-attr:facsimileTelephoneNumber writable-attr:roomNumber \
writable-attr:userPassword
```

These definitions apply only in the context of the *join* view. By default all attributes can be read and written if you access the LDAP data view directly.

5 Define the set of attributes that can be read from and written to the secondary data view through a join data view.

```
% dpconf set-jdbc-data-view-prop mysql1-view viewable-attr:dn \
 viewable-attr:objectclass viewable-attr:sn viewable-attr:roomNumber \
 viewable-attr:userpassword viewable-attr:jobtitle viewable-attr:countryName \
 viewable-attr:telephoneNumber
% dpconf set-jdbc-data-view-prop mysql1-view writable-attr:dn \
writable-attr:objectclass writable-attr:sn writable-attr:roomNumber \
writable-attr:userpassword writable-attr:jobtitle \
writable-attr:countryName writable-attr:telephoneNumber
```

These definitions apply only in the context of the *join* view. By default all attributes can be read and written if you access the JDBC data view directly.

▼ To Create the Required ACIs

1 As the proxy manager, add a global ACI that allows anonymous access to the join data view.

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=myjoin1,cn=virtual access controls
changetype: add
```

```
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=join")
(version 3.0; acl "anonymous_access"; allow(all) userdn="ldap:///anyone";)
cn: myjoin1
```

2 **Configure the connection handler to use the pool of ACIs added previously.**

```
% dpconf set-connection-handler-prop default-connection-handler aci-source:myjoin1
```

▼ To Test the Join Data View

1 **As an anonymous user, search the join data view.**

In this step, we search `Kirsten Vaughan`'s entry to see whether data from both join views is retrieved.

```
% ldapsearch -p 1389 -b o=join "uid=kvaughan"
```

Note that the returned entry includes the attributes from both the LDAP data view and the JDBC data view.

2 **As a user under `o=join`, modify the `userPassword` attribute to verify that you can write to the join data view.**

```
% ldapmodify -p 1389
dn: uid=kvaughan,ou=people,o=join
changetype: modify
replace: userPassword
userPassword: myPassword
```

Joining Multiple Disparate Data Sources

This configuration describes an organization, Example.com, whose specific directory service requirements are met by some of the features of a virtual directory.

Data Storage Scenario

Example.com stores organizational data in multiple disparate data sources. For legacy reasons, user data is spread across an LDAP directory, a flat LDIF file, and an SQL database. The HR department stores user data in an LDAP directory, with a base DN of `o=example.com`. The Payroll department stores data in an SQL database. Administrative data such as departments and building numbers is stored by the administration department in an LDIF file, with a base DN of `dc=example,dc=com`.

In addition, Example.com has acquired a company named Company22. Company 22 also stores its user data in an LDAP directory, with a base DN of `dc=company22,dc=com`.

The following diagram provides a high level view of how Example.com's user data is stored.

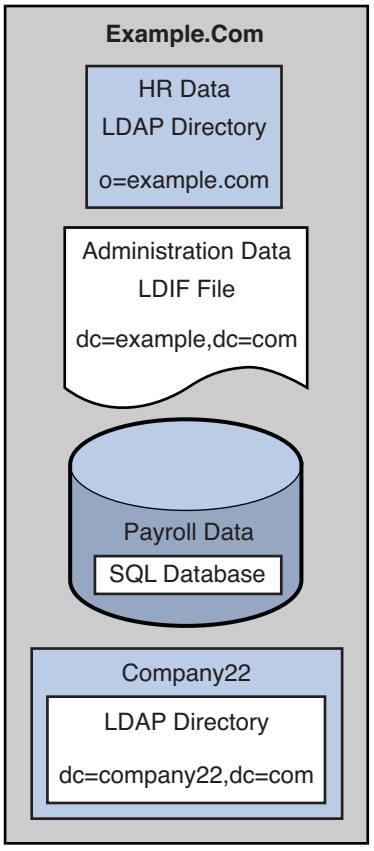


FIGURE 22–2 Data Storage In Disparate Sources

Client Application Requirements

Example.com has several LDAP client applications that require access to the data stored in the disparate data sources. The requirements of the client applications are not all the same. Different views of the data are required. In some cases, the clients require the data to be aggregated. In addition, some client applications require access to Company22's user data so that these new employees of Example.com can be administered along with the old employees.

The following diagram provides a high level view of Example.com's client application requirements.

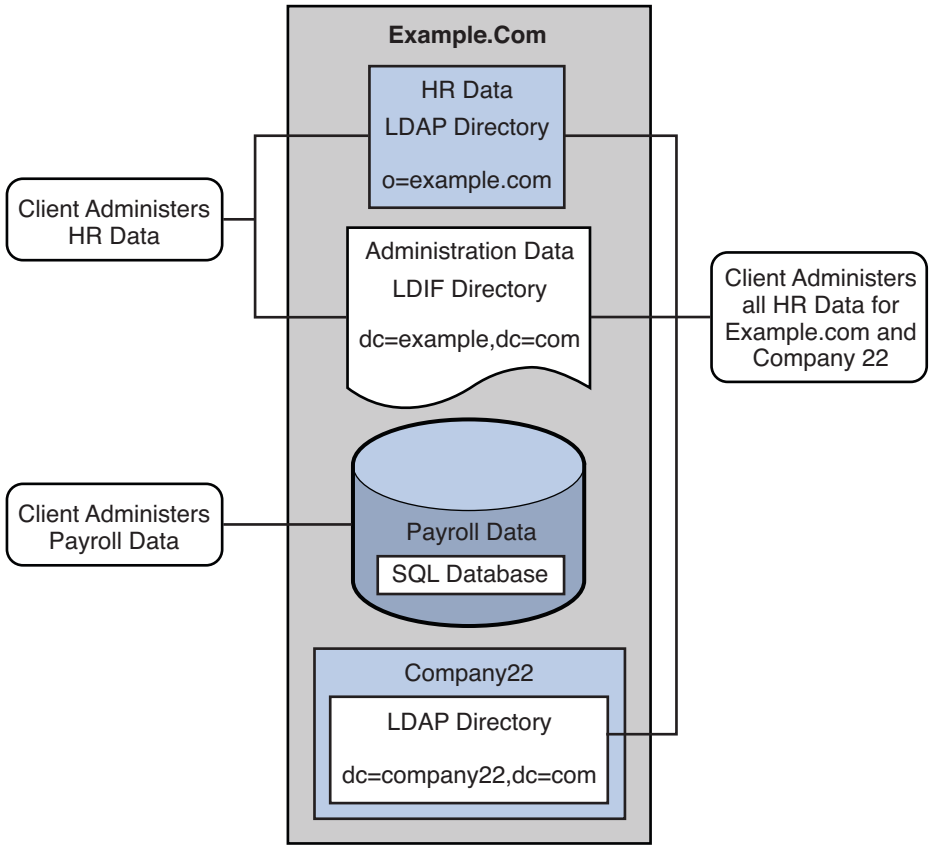


FIGURE 22–3 Client Application Requirements

The following sections walk you through sufficient configuration Directory Proxy Server data views to meet the client application requirements described in this sample scenario. For information about how data views work, see Chapter 17, "Directory Proxy Server Distribution," in *Sun Directory Server Enterprise Edition 7.0 Reference* and Chapter 18, "Directory Proxy Server Virtualization," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The configuration of the sample scenario is divided into the following sections:

- "Aggregate Data From the HR LDAP Directory and the Administration LDIF File" on page 464
- "Add Data From Company 22 to Example.Com's DIT by Renaming the DN" on page 466
- "Add Company 22's Data to the HR Data" on page 468
- "Enable LDAP Clients to Access the Payroll Data in an SQL Database" on page 469
- "Add Virtual Access Control" on page 471

Aggregate Data From the HR LDAP Directory and the Administration LDIF File

The HR department stores information such as employee names, job start data, and job level. The administration department stores additional data such as building codes and office numbers. The client application that handles the HR data requires access to the combined data from both sources. Both data sources have a common attribute, the `employeeNumber` that exists in each entry.

The following diagram illustrates the requirements of the client application.

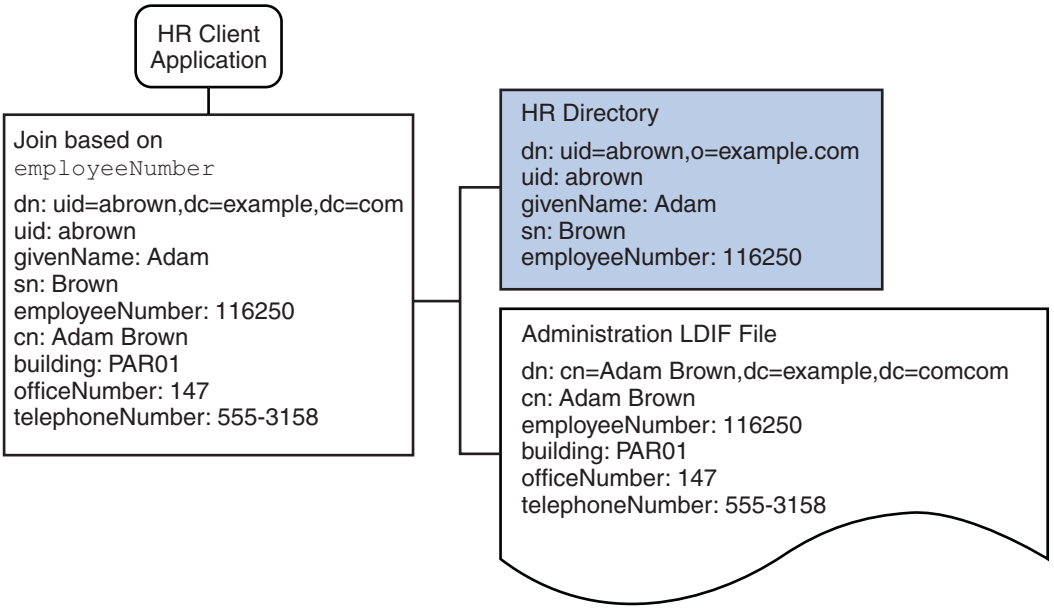


FIGURE 22–4 Aggregation of Data From LDAP Directory and LDIF File

To fulfill this application requirement, a data view is created for the payroll directory and for the administration LDIF file. These two data views are then joined to provide access to the aggregated data. This common attribute enables Directory Proxy Server to aggregate the data for each user.

For simplicity, the commands used in this section assume the following information:

- A Directory Proxy Server instance runs on the local host, with the default LDAP port (`389`).
- The Directory Proxy Server instance is located at `/local/myDPS`.
- The path to the file containing the Proxy Manager password has been set as a variable, `LDAP_ADMIN_PWF`.

- The payroll LDAP directory runs on a host named `payrollHost`, on port `2389`.
- The LDIF file used to store the administration data is named `example.ldif`.

To obtain the complete syntax of each command, run the command without any options. For example:

```
$ dpconf create-ldap-data-view
Operands are missing
Usage: dpconf create-ldap-data-view VIEW_NAME POOL_NAME SUFFIX_DN
```

Create and Enable an LDAP Data View for the Payroll Directory

1. **Create an LDAP data source for the payroll directory.**

   ```
   $ dpconf create-ldap-data-source payroll-directory payrollHost:2389
   ```

2. **Create an LDAP data source pool for the payroll directory.**

   ```
   $ dpconf create-ldap-data-source-pool payroll-pool
   ```

3. **Attach the payroll data source to the data source pool.**

   ```
   $ dpconf attach-ldap-data-source payroll-pool payroll-directory
   ```

4. **Configure the weights of the attached data source.**

   ```
   $ dpconf set-attached-ldap-data-source-prop -h payrollHost -p 2389 \
   payroll-pool payroll-directory add-weight:2 \
   bind-weight:2 compare-weight:2 delete-weight:2 \
   modify-dn-weight:2 modify-weight:2 search-weight:2
   ```

5. **Create an LDAP data view for the payroll directory.**

   ```
   $ dpconf create-ldap-data-view payroll-view payroll-pool o=example.com
   ```

6. **Enable the LDAP data view so that client requests can be routed to this data view.**

   ```
   $ dpconf set-ldap-data-view-prop payroll-view is-enabled:true
   ```

7. **Restart Directory Proxy Server for the changes to take effect.**

   ```
   $ dpadm restart /local/myDPS
   ```

Create and Enable an LDIF Data View for the Administration Data

1. **Create an LDIF data view for the administration data.**

   ```
   $ dpconf create-ldif-data-view admin-view example.ldif dc=example,dc=com
   ```

2. **Enable the LDIF data view for the administration data.**

```
$ dpconf set-ldif-data-view-prop admin-view is-enabled:true
```

3. **Specify that the administrator view contains entries that are used by more than one entry in the payroll view.**

```
$ dpconf set-ldif-data-view-prop admin-view contains-shared-entries:true
```

When this property is set to `TRUE`, deleting an entry in the payroll data view will not result in the deletion of the shared entry in the administrator data view. Adding an entry to the payroll data view will only add the entry to the secondary data view if it does not already exist.

4. **Restart Directory Proxy Server for the changes to take effect.**

```
$ dpadm restart /local/myDPS
```

▼ Join the Payroll Data View and the Administrator Data View

1. **Create a filter join rule on the administrator data view that specifies how the data should be aggregated.**

The following join rule specifies that data should be joined based on the `employeeNumber` attribute of the user entry.

```
$ dpconf set-ldif-data-view-prop admin-view \
filter-join-rule:employeeNumber=\${payroll-view.employeeNumber}
```

2. **Create a join data view that aggregates the two data views.**

For the join data view, the organization uses the suffix DN `dc=example,dc=com`.

```
$ dpconf create-join-data-view example-join-view payroll-view admin-view \
dc=example,dc=com
```

Add Data From Company 22 to Example.Com's DIT by Renaming the DN

The user data for Company 22 is stored under the DN `dc=company22,dc=com`. While Example.com wants to keep this user data separate in most cases, one client application needs to administer Company 22 employees along with the rest of the Example.com employees. This client application requires Company 22's user data to *look like* Example.com data.

The following diagram illustrates the requirements of the client application.

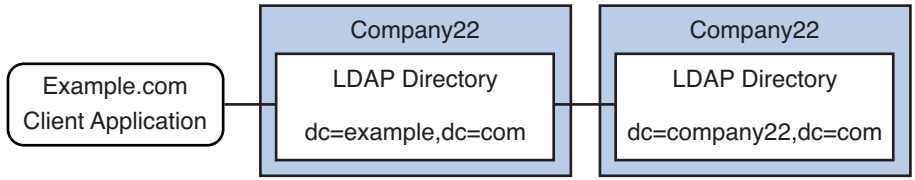


FIGURE 22–5 DN Renaming

To fulfill this application requirement, a data view with a virtual DN of `dc=example,dc=com` is created for the Company 22's directory.

For simplicity, the commands used in this section assume the following information:

- A Directory Proxy Server instance runs on the local host, with the default LDAP port (`389`).
- The Directory Proxy Server instance is located at `/local/myDPS`.
- The path to the file containing the Proxy Manager password has been set as a variable, `LDAP_ADMIN_PWF`.
- The Company 22 LDAP directory runs on a host named `company22Host`, on port `2389`.

▼ Create a Data View For Company 22's Directory With a Virtual DN

1 Create an LDAP data source for Company 22's directory.

```
$ dpconf create-ldap-data-source company22-directory company22Host:2389
```

2 Create an LDAP data source pool for Company 22's directory.

```
$ dpconf create-ldap-data-source-pool company22-pool
```

3 Attach Company 22's data source to the data source pool.

```
$ dpconf attach-ldap-data-source company22-pool company22-directory
```

4 Configure the weights of the attached data source.

```
$ dpconf set-attached-ldap-data-source-prop -p 2389 \
company22-pool company22-directory add-weight:2 \
bind-weight:2 compare-weight:2 delete-weight:2 \
modify-dn-weight:2 modify-weight:2 search-weight:2
```

5 Create an LDAP data view for Company 22's directory with a virtual DN of `dc=example,dc=com`.

```
$ dpconf create-ldap-data-view company22-view company22-pool dc=example,dc=com
```

6 Instruct Directory Proxy Server to map this virtual DN to the real DN that is in Company 22's directory.

```
$ dpconf set-ldap-data-view-prop company22-view \
dn-mapping-source-base-dn:dc=company22,dc=com
```

7 **Enable the LDAP data view for Company 22's directory so that client requests can be routed to this data view.**

```
$ dpconf set-ldap-data-view-prop company22-view is-enabled:true
```

8 **Restart Directory Proxy Server for the changes to take effect.**

```
$ dpadm restart /local/myDPS
```

Add Company 22's Data to the HR Data

The HR department requires an aggregated view of the HR data for Example.com and the newly acquired Company 22. The following diagram illustrates the requirements of the global HR application.

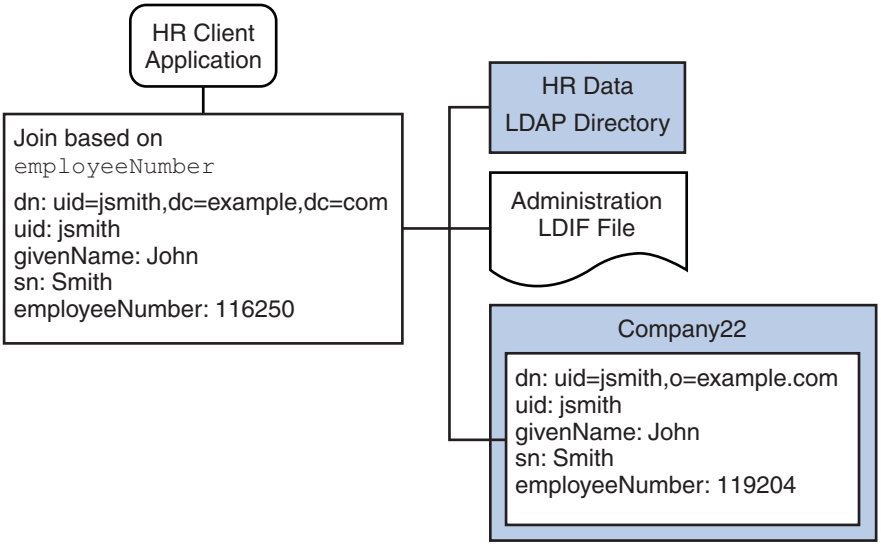


FIGURE 22–6 Aggregation of Data From Join Data View and LDAP Data View

▼ Join the Example Join Data View and the Company 22 Data View

1 **Create a filter join rule on the Company 22 data view that specifies how the data should be aggregated.**

The following join rule specifies that data should be joined based on the `employeeNumber` attribute of the user entry.

```
$ dpconf set-ldif-data-view-prop company22-view \
filter-join-rule:employeeNumber=\${example-join-view.employeeNumber}
```

2 Create a join data view that aggregates Company 22's data view and Example.com's join data view.

```
$ dpconf create-join-data-view global-join-view example-join-view \
company22-view dc=example,dc=com
```

Enable LDAP Clients to Access the Payroll Data in an SQL Database

Example.com's payroll department stores salary data in an SQL database. The database has two tables, and an `employee` table and a `salary` table. Example.com has an LDAP client application that requires access to that data. The client application requires the SQL data to *look like* LDAP data.

The following diagram illustrates the requirements of the client application.

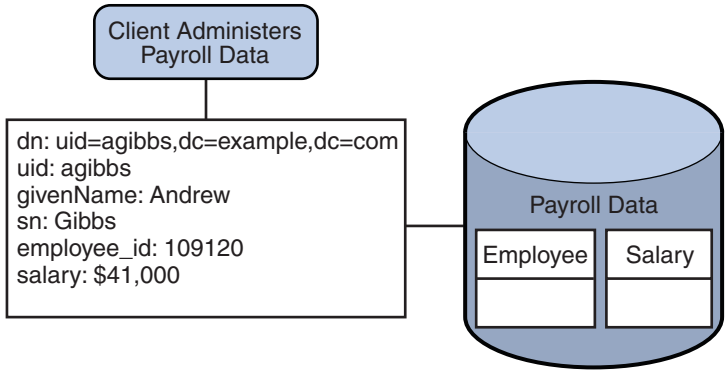


FIGURE 22–7 JDBC Dataview Providing Access to an SQL Database

To fulfill this application requirement, a JDBC data view is created that maps columns in the SQL tables to LDAP attributes.

For simplicity, the commands used in this section assume the following information:

- A Directory Proxy Server instance runs on the local host, with the default LDAP port (`389`).
- The Directory Proxy Server instance is located at `/local/myDPS`.
- The path to the file containing the Proxy Manager password has been set as a variable, `LDAP_ADMIN_PWF`.
- The SQL database is up and running.
- The `JAVA_HOME` variable has been set to the correct Java path.
- The password to the SQL database is `myPassword` stored in the `myPasswordFile` file.

▼ Create a JDBC Data View For Example.com's Payroll Database

1 Create a JDBC data source for the payroll database.

```
$ dpconf create-jdbc-data-source -b payrollsqldb \
  -B jdbc:payrollsqldb:payrollsql://localhost/ \
  -J file://payrollsqldb.jar \
  -S org.payrollsqldb.jdbcDriver payroll-src
```

2 Configure the JDBC data source with the properties of the SQL database.

```
$ dpconf set-jdbc-data-source-prop payroll-src \
  db-user:proxy db-pwd-file:password-file-location/myPasswordFile
```

3 Enable the JDBC data source.

```
$ dpconf set-jdbc-data-source-prop payroll-src is-enabled:true
```

4 Create a JDBC data source pool for the payroll database.

```
$ dpconf create-jdbc-data-source-pool payroll-pool
```

5 Attach the payroll data source to the data source pool.

```
$ dpconf attach-jdbc-data-source payroll-pool payroll-src
```

6 Create a JDBC data view for the payroll database, with a virtual DN of `o=payroll`.

```
$ dpconf create-jdbc-data-view payroll-view payroll-pool o=payroll
```

7 Create a JDBC table for each table in the SQL database.

```
$ dpconf create-jdbc-table jdbc-employee employee
$ dpconf create-jdbc-table jdbc-salary salary
```

8 Add a JDBC attribute for each column in the SQL tables.

```
$ dpconf add-jdbc-attr jdbc-employee eid employee_id
$ dpconf add-jdbc-attr jdbc-employee first firstname
$ dpconf add-jdbc-attr jdbc-employee last lastname
$ dpconf add-jdbc-attr jdbc-employee description description
$ dpconf add-jdbc-attr jdbc-employee spouse spousename
$ dpconf add-jdbc-attr jdbc-salary salary salary
$ dpconf add-jdbc-attr jdbc-salary social ssn
```

9 Specify which attributes can be viewed and which can be written, through the JDBC data view.

```
$ dpconf set-jdbc-data-view-prop payroll-view \
 viewable-attr:eid \
 viewable-attr:first \
 viewable-attr:last \
 viewable-attr:desc \
 viewable-attr:spouse \
```

```
 viewable-attr:salary \
 viewable-attr:social
 $ dpconf set-jdbc-data-view-prop payroll-view \
 writable-attr:eid \
 writable-attr:first \
 writable-attr:last \
 writable-attr:description \
 writable-attr:spouse \
 writable-attr:salary \
writable-attr:social
```

10 Create a JDBC object class that maps to an LDAP object class.

The following command creates an object class that maps to the LDAP `person` object class. The object class specifies that the employee table should be used as the primary table, and that the salary table should be used as the secondary table. The `eid` attribute should be used to construct the DN.

```
$ dpconf create-jdbc-object-class payroll-view \
 person jdbc-employee jdbc-salary eid
```

11 Create a filter join rule on the secondary table that specifies how data from the secondary table should be linked to data from the primary table.

The following join rule specifies that data should be joined based on the `employee_id` attribute.

```
$ dpconf set-jdbc-table-prop jdbc-salary \
filter-join-rule:employee_id=\${employee.employee_id}
```

12 Create a super class on the JDBC object class.

```
$ dpconf set-jdbc-object-class-prop payroll-view person super-class:extensibleObject
```

Add Virtual Access Control

Access control on LDAP directories is handled by defining ACIs in the directories themselves. When data sources are accessed through virtual data views, ACIs must be defined that apply only to the data viewed through these data views.

Any access that goes through Directory Proxy Server is controlled by a *connection handler*. For information about connection handlers, see Chapter 25, “Connections Between Clients and Directory Proxy Server .”

▼ Add an ACI That Allows Anonymous Access

1 Add the ACI.

```
$ ldapadd -v -D "cn=proxy manager" -w password -p 389
dn: cn=ldifonly-acis,cn=virtual access controls
objectclass: top
```

```
objectclass: aciSource
cn: ldifonly-acis
dpsaci: (targetattr="*")(version 3.0; acl "anonymous_access"; allow(all) \
(userdn="ldap:///anyone");)
```

2 Point the connection handler to the virtual ACI.

```
$ dpconf set-connection-handler-prop anonymous aci-source:ldifonly-acis
```

3 Enable the connection handler.

```
$ dpconf set-connection-handler-prop anonymous is-enabled:true
```

◆ ◆ ◆ CHAPTER 23

Virtual Data Transformations

Virtual data transformations are defined on existing data views, and enables you to create virtual data using the virtual data views. For information about how they work, see "Virtual Data Transformations" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Configuring Virtual Data Transformations

You can add a virtual data transformation to any type of data view: an LDAP data view, an LDIF data view, a join data view, or a JDBC data view.

▼ To Add a Virtual Transformation

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 Add the transformation to a data view.

```
$ dpconf add-virtual-transformation -h host -p port view-name \
 transformation-model transformation-action attribute-name [parameters...]
```

The *transformation-model* can be one of the `mapping`, `write`, and `read` transformations.

The *transformation-action* can be one of the `add-attr`, `remove-attr`, `add-attr-value`, `remove-attr-value`, `def-value`, and `attr-value-mapping` actions.

Note that *parameters* might be mandatory, depending on the *transformation-model* and the *transformation-action*.

For information about transformation models, transformation actions, and transformation parameters, see "Virtual Data Transformations" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

2. **(Optional) View the list of virtual transformations that are defined on a data view.**

```
$ dpconf list-virtual-transformations -h host -p port view-name
```

▼ To Remove a Virtual Transformation

- **Remove the virtual transformation using the following command.**

```
dpconf remove-virtual-transformation view_name transformation_name
```

Virtual Transformation Examples

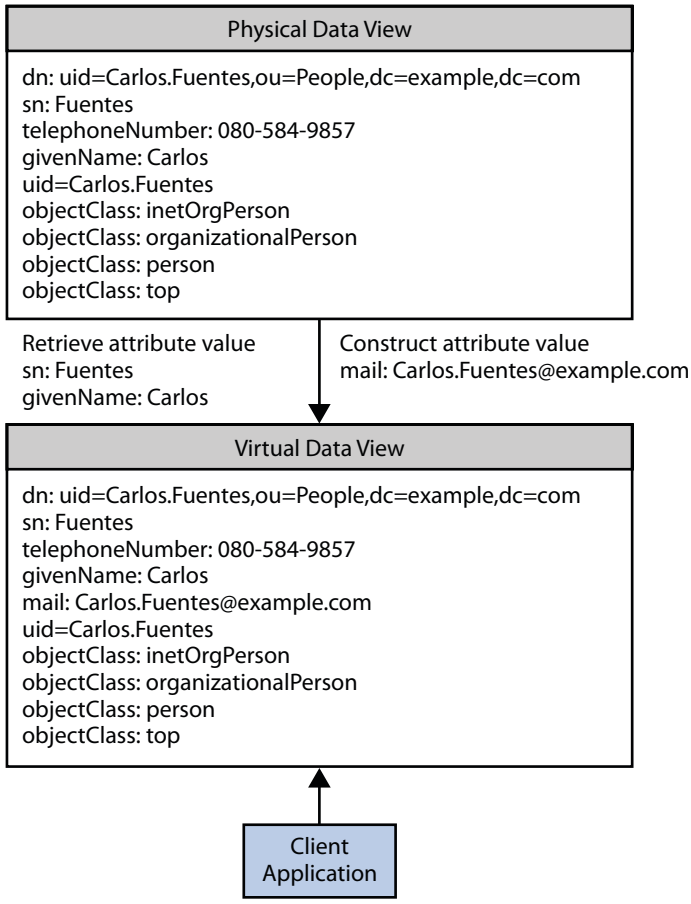
The following sections provide use cases in which virtual data views are required, and the combination of transformation models and actions required to implement the use cases.

Deriving an Attribute From the Existing Attributes of an Entry

Use the following transformation rule to derive an attribute from the existing attributes of an entry. For example, when the following transformation rule is applied, it displays the `mail` attribute derived from the `givenName` and `sn` attributes.

```
$ dpconf add-virtual-transformation dataview1 read add-attr \
mail \${givenName}.\${sn}@example.com
```

The following diagram indicates the transformation that occurs on user entries when they are returned in a search.



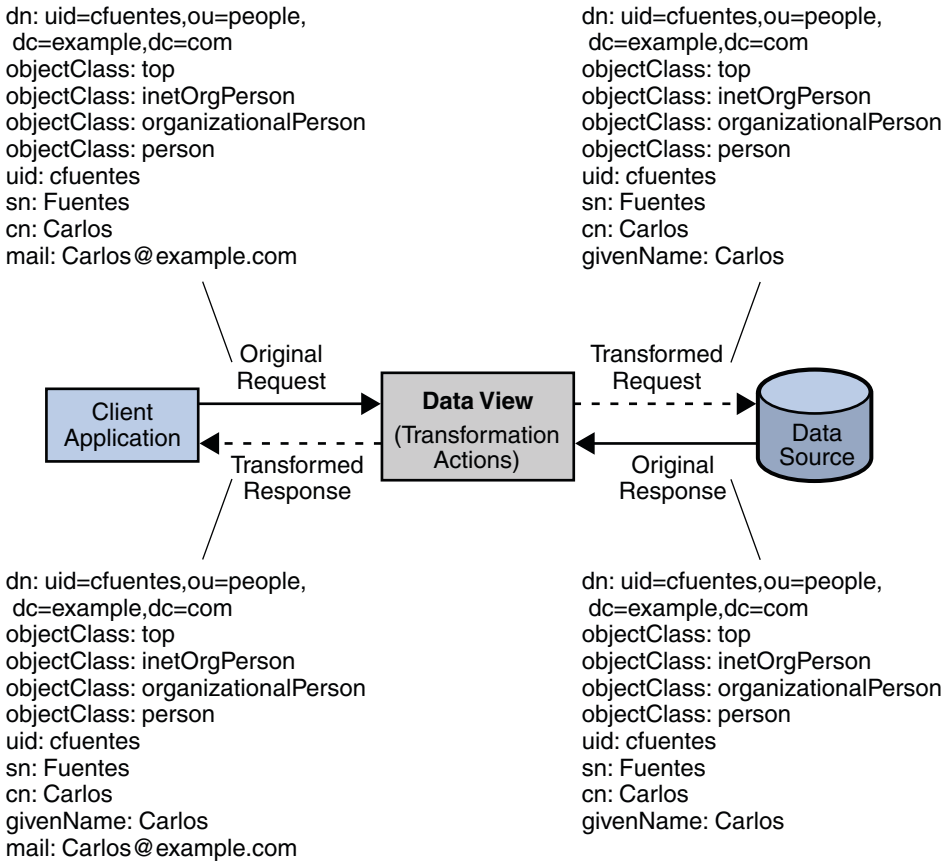
Mapping a Virtual Attribute to a Physical Attribute

Use the following mapping transformation rule to add an attribute that is delivered as a part of a pure virtual attribute. For example, when the following transformation rule is applied, the `givenName` is stored in the server even when it is not specified in the entry. The value is taken from the pure virtual attribute which is defined as `mail \${givenName}@example.com`.

```
$ dpconf add-virtual-transformation dataview1 mapping add-attr \
mail \${givenName}@example.com
```

First add an entry that contains a virtual attribute, `mail`, but no `givenName` attribute. The virtual transformation generates the value for the `givenName` attribute and the entry is stored with `givenName` but without the `mail` attribute. Then, doing a search on using the `uid` attribute, retrieve the value for `givenName`, and the same virtual transformation generates the value of the virtual attribute `mail`.

The following diagram indicates the transformation that occurs on user entries.

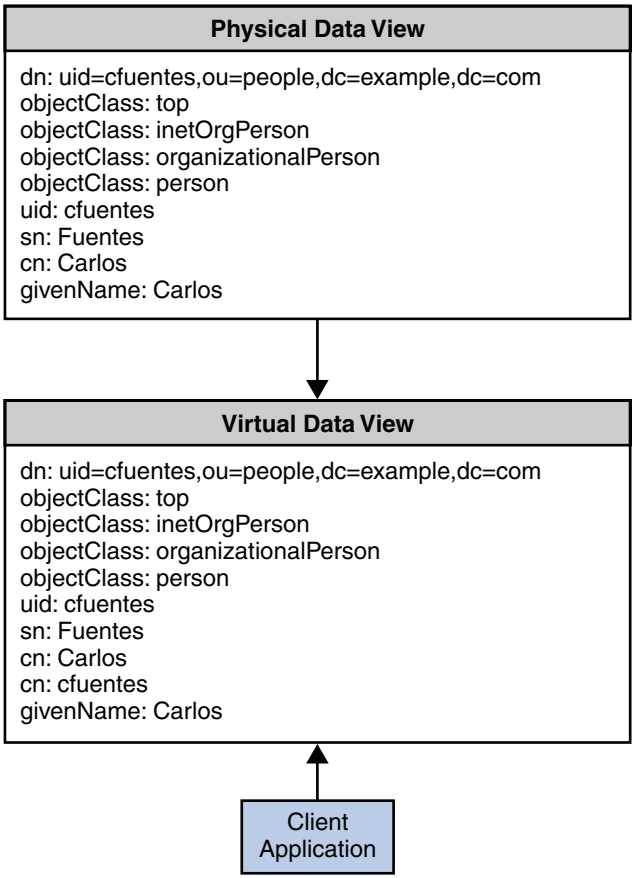


Displaying a Second Virtual Value of an Attribute, Specified by Another Physical Attribute

Use the following transformation to display the value of an attribute that is specified by another attribute. For example, displaying `uid` as `cn` along with the value of `cn` that is already stored in the entry. The following does not store the additional value to `cn` but before the result is returned to the client, the transformation is applied.

```
$ dpconf add-virtual-transformation dataview1 read add-attr-value cn \${uid}
```

The following diagram indicates the transformation that occurs on user entries when they are returned in a search.

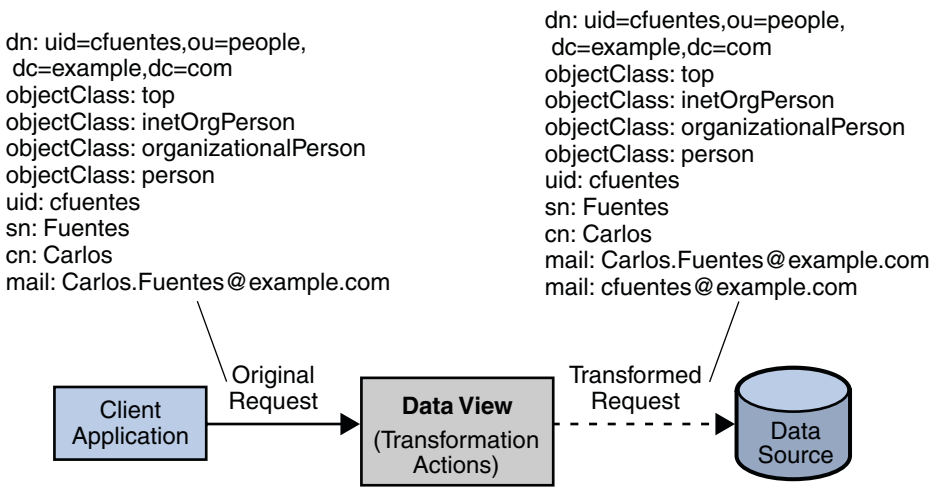


Storing a Second Value to an Attribute Specified by Another Physical Attribute

Use the following transformation rule to store the value of an attribute along with the value that you provide while adding a new entry. In this scenario, when you add an entry, an additional value for the `mail` attribute is stored. This transformation is applied only at the time of creating new entries.

```
$ dpconf add-virtual-transformation dataview1 write add-attr-value \
mail \${uid}@example.com
```

The following diagram indicates the transformation that occurs on an add request.

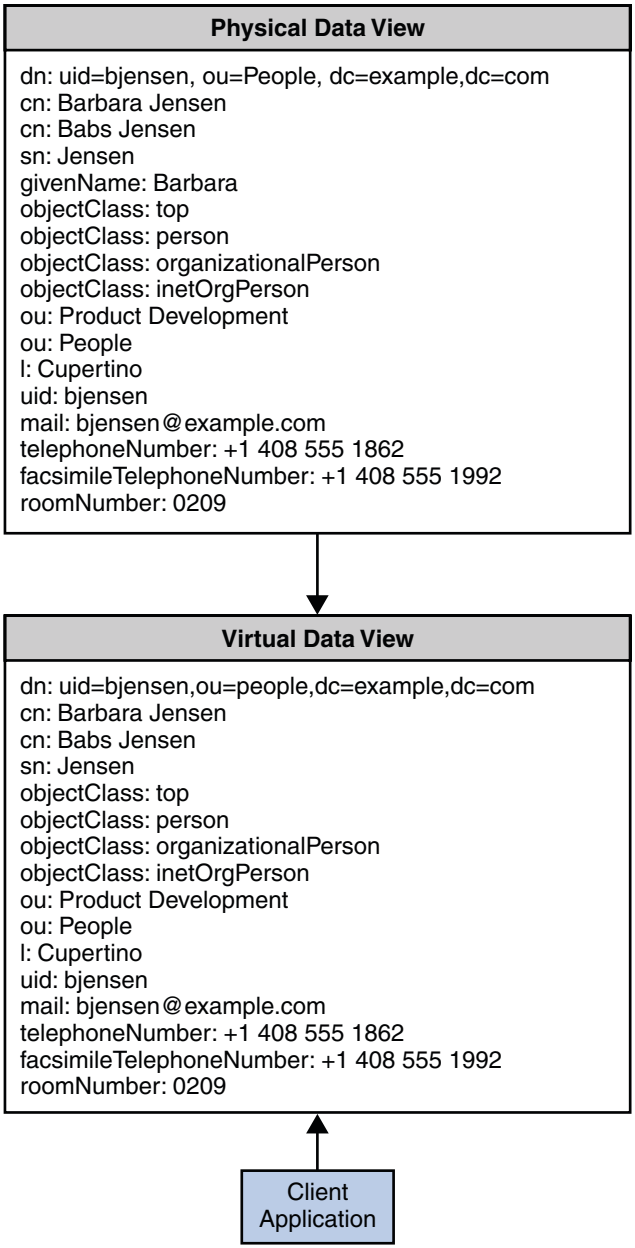


Removing an Attribute From the Output

Use the following transformation rule if you do not want to display an attribute in the output. For example, when the following transformation rule is applied, the `givenName` is not returned in the output.

```
dpconf add-virtual-transformation dataview1 read remove-attr givenName
```

The following diagram indicates the transformation that occurs on user entries when they are returned in a search.

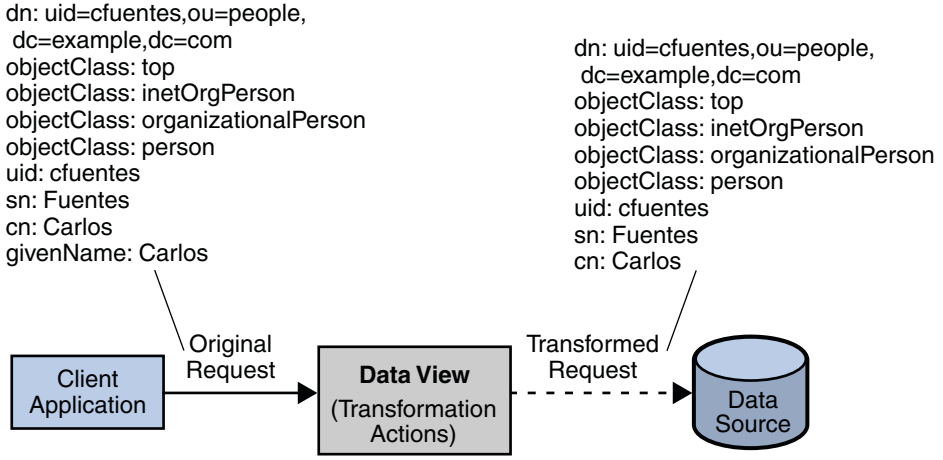

Physical Data View
dn: uid=bjensen, ou=People, dc=example,dc=com
cn: Barbara Jensen
cn: Babs Jensen
sn: Jensen
givenName: Barbara
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Product Development
ou: People
l: Cupertino
uid: bjensen
mail: bjensen@example.com
telephoneNumber: +1 408 555 1862
facsimileTelephoneNumber: +1 408 555 1992
roomNumber: 0209
Virtual Data View
dn: uid=bjensen,ou=people,dc=example,dc=com
cn: Barbara Jensen
cn: Babs Jensen
sn: Jensen
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Product Development
ou: People
l: Cupertino
uid: bjensen
mail: bjensen@example.com
telephoneNumber: +1 408 555 1862
facsimileTelephoneNumber: +1 408 555 1992
roomNumber: 0209
Client Application

Masking an Attribute While Saving an Entry

Use the following transformation rule if you do not want to store a specific attribute. For example, when the following transformation rule is applied, the `givenName` attribute is not stored in the physical database. This transformation is applied only at the time of creating new entries.

```
$ dpconf add-virtual-transformation dataview1 write remove-attr givenName
```

The following diagram indicates the transformation that occurs on an add request.

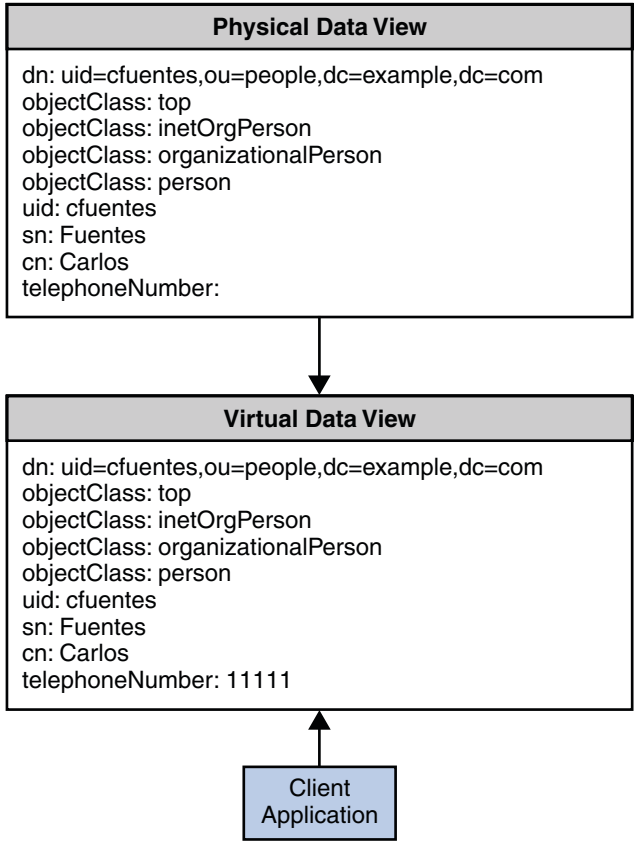


Displaying a Default Value of an Attribute

Use the following transformation if you want to display a default value assigned to an attribute. For example, when the following transformation is applied, a default telephone number in the entries that do not contain their own telephone number is displayed.

```
$ dpconf add-virtual-transformation data-view read 11111 telephoneNumber default-number
```

The following diagram indicates the transformation that occurs on user entries when they are returned in a search.

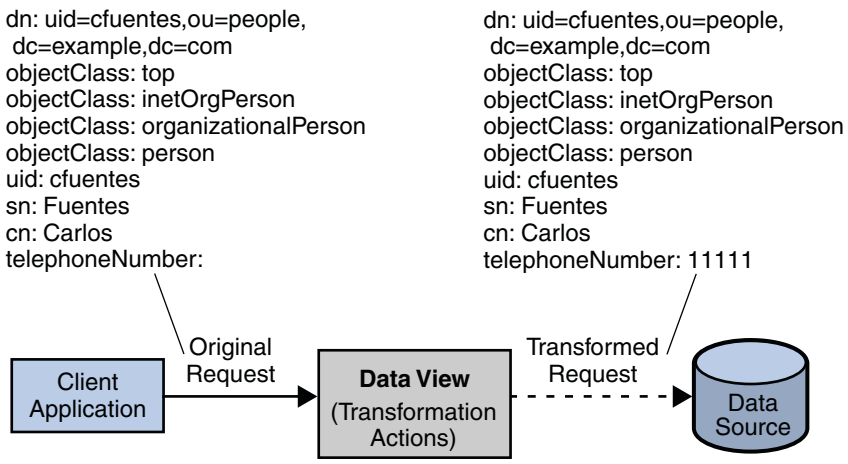


Storing a Default Value to an Attribute

The default value is stored only if the value for an attribute is not specified during the creation of an entry. Use the following transformation rule if you want to store an attribute with the default value. For example, when the following transformation is applied, a default telephone number with each entry you create is added. This transformation is applied only at the time of adding an entry.

```
$ dpconf add-virtual-transformation dataview1 write 11111 \
telephoneNumber telephone-number
```

The following diagram indicates the transformation that occurs on an add request.


dn: uid=cfuentes,ou=people,
dc=example,dc=com
objectClass: top
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
uid: cfuentes
sn: Fuentes
cn: Carlos
telephoneNumber:
dn: uid=cfuentes,ou=people,
dc=example,dc=com
objectClass: top
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
uid: cfuentes
sn: Fuentes
cn: Carlos
telephoneNumber: 11111
Original
Request
Transformed
Request
Client
Application
Data View
(Transformation
Actions)
Data
Source

◆ ◆ ◆ CHAPTER 24

Connections Between Directory Proxy Server and Back-End LDAP Servers

This chapter describes how to configure connections between Directory Proxy Server and back-end LDAP servers. The chapter covers the following topics:

Configuring Connections Between Directory Proxy Server and Back-End LDAP Servers

When an LDAP data source is created, the number of default connections opened for the LDAP data source are six, that is, two for each read, bind, and write operations respectively. To verify the default connections, type the following command:

```
$ dpconf get-ldap-data-source-prop src-name num-read-init num-write-init num-bind-init
num-bind-init   :  2
num-read-init   :  2
num-write-init  :  2
```

The number of connections increases automatically when the traffic increases.

For information about how to configure connections between Directory Proxy Server and back-end LDAP servers, see the following procedures:

▼ To Configure the Number of Connections Between Directory Proxy Server and Back-End LDAP Servers

Note – This procedure configures the number of connections for bind operations. To configure the number of connections for read or write operations, perform the same procedure but replace `bind` with `read` or `write`.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Configure the initial number of connections between Directory Proxy Server and a back-end LDAP server for bind operations.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-init:new-value
   ```

2. **Configure the increment of connections for bind operations.**

   The increment is the number of connections that are added each time more than the current number of connections are requested.

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-incr:new-value
   ```

3. **Configure the maximum number of connections for bind operations.**

   When this maximum number of connections is reached, no more connections can be added.

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-limit:new-value
   ```

▼ To Configure Connection Timeout

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure the maximum length of time that Directory Proxy Server can attempt to connect to a data source.**

  ```
  $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
   connect-timeout:new-value
  ```

  For example, configure the connection timeout to 10 milliseconds.

  ```
  $ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name connect-timeout:10
  ```

▼ To Configure Connection Pool Wait Timeout

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure the maximum length of time that Directory Proxy Server can wait for an established connection in a connection pool to become available.**

```
$ dpconf set-server-prop -h host -p port data-source-name \
 connection-pool-wait-timeout:value
```

For example, configure the timeout to 20 seconds.

```
$ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name \
 connection-pool-wait-timeout:20000
```

Configuring SSL Between Directory Proxy Server and Back-End LDAP Servers

The following procedure describes how to configure SSL between Directory Proxy Server and back-end LDAP servers.

▼ To Configure SSL Between Directory Proxy Server and a Back-End LDAP Server

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Configure a secure port between Directory Proxy Server and the back-end LDAP server.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 ldaps-port:port-number
```

2. **Configure when SSL is used for connections between Directory Proxy Server and the back-end LDAP server.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name ssl-policy:value
```

   - If *value* is `always`, SSL is always used for connections.
   - If *value* is `client`, SSL is used if the client is using SSL.

   If the connection is not using SSL, you can promote the connection to SSL by using the `startTLS` command.

Transport Layer Security (TLS) is the standard version of SSL. TLS over LDAP is the IETF approved standard way of securing LDAP. LDAPS is a de facto standard but leads to some complexity such as having two ports instead of only one port for the service.

3. **Choose the protocols and ciphers for SSL as described in "Choosing SSL Ciphers and SSL Protocols for Directory Proxy Server" on page 486.**

4. **Configure Directory Proxy Server to validate an SSL server certificate from the back-end LDAP server.**

   For information, see "To Add a Certificate From a Back-End Directory Server to the Certificate Database on Directory Proxy Server" on page 400.

5. **If the back-end LDAP server requests a certificate from Directory Proxy Server, configure Directory Proxy Server to send an SSL client certificate.**

   For information, see "Exporting a Certificate to a Back-End LDAP Server" on page 401.

6. **Restart the instance of Directory Proxy Server for the changes to take effect.**

   For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Choosing SSL Ciphers and SSL Protocols for Directory Proxy Server

The ciphers and protocols that can be used by Directory Proxy Server depend on the Java™ Virtual Machine (JVM™) that is being used. By default, Directory Proxy Server uses the default ciphers and protocols that are enabled for the JVM machine.

▼ To Choose the List of Ciphers and Protocols

Use this procedure to retrieve the supported ciphers and protocols, and the enabled ciphers and protocols. If a cipher or protocol is supported, you can enable or disable it.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **View the list of supported ciphers and protocols.**

```
$ dpconf get-server-prop -h host -p port supported-ssl-cipher-suites \
 supported-ssl-protocols
```

2 **View the list of enabled ciphers and protocols.**

```
$ dpconf get-server-prop -h host -p port enabled-ssl-cipher-suites \
 enabled-ssl-protocols
```

3 **Enable one or more supported ciphers or protocols.**

a. **Enable one or more supported ciphers.**

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-suites:supported-ssl-cipher-suite \
 [enabled-ssl-cipher-suites:supported-ssl-cipher-suite ...]
```

To add a cipher to an existing list of supported ciphers, use this command:

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-suites+:supported-ssl-cipher-suite
```

b. **Enable one or more supported protocols.**

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol \
 [enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol ...]
```

To add a protocol to an existing list of supported protocols, use this command:

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols+:supported-ssl-cipher-protocol
```

4 **(Optional) Disable a supported cipher or protocol.**

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols-:supported-ssl-cipher-protocol
```

Forwarding Requests to Back-End LDAP Servers

This section contains information about the various methods you can use to forward requests from Directory Proxy Server to back-end LDAP servers.

Forwarding Requests With Bind Replay

For information about bind replay for client credentials in Directory Proxy Server, see "Directory Proxy Server Configured for BIND Replay" in *Sun Directory Server Enterprise Edition 7.0 Reference*. The following procedure describes how to forward requests from Directory Proxy Server to a back-end LDAP server by using bind replay.

▼ To Forward Requests With Bind Replay

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure the data source client credentials to authenticate to a back-end LDAP server by using the credentials provided by a client.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-client-identity
```

Forwarding Requests With Proxy Authorization

For information about proxy authorization in Directory Proxy Server, see "Directory Proxy Server Configured for Proxy Authorization" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This section contains procedures for forwarding requests by using proxy authorization and by using a proxy authorization control.

▼ To Forward Requests by Using Proxy Authorization

1. **Configure the data source to expect proxy authorization controls of either version 1 or version 2.**

   For example, configure the data source to expect proxy authorization controls of version 1.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-use-v1:true
```

   Alternatively, configure the data source to expect proxy authorization controls of version 2.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-use-v1:false
```

2. **Configure the data source to authenticate to a back-end LDAP server by using proxy authorization.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-proxy-auth
```

   To configure a data source to authenticate to a back-end LDAP server by using proxy authorization for write operations only, run this command:

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-proxy-auth-for-write
```

When write operations only are performed with a proxy authorization control, the client identity is not forwarded to the LDAP server for read requests. For more information about forwarding requests without the client identity, see "Forwarding Requests Without the Client Identity" on page 489.

3 **Configure the data source with the bind credentials of Directory Proxy Server.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 bind-dn:DPS-bind-dn bind-pwd-file:filename
```

4 **Configure the data source with the timeout.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-check-timeout:value
```

Directory Proxy Server verifies that the client DN has the relevant ACIs for proxy authorization by using the `getEffectiveRights` command. The result is cached in Directory Proxy Server and renewed when the `proxied-auth-check-timeout` expires.

5 **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

▼ To Forward Requests by Using Proxy Authorization When the Request Contains a Proxy Authorization Control

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure Directory Proxy Server to accept proxy authorization controls of version 1, version 2, or both.**

```
$ dpconf set-server-prop -h host -p port allowed-ldap-controls:proxy-auth-v1 \
 allowed-ldap-controls:proxy-auth-v2
```

Forwarding Requests Without the Client Identity

The following procedure describes how to forward requests from Directory Proxy Server to a back-end LDAP server without forwarding the client identity.

▼ To Forward Requests Without the Client Identity

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Configure the data source to authenticate to a back-end LDAP server by using the credentials of Directory Proxy Server.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    client-cred-mode:use-specific-identity
   ```

2. **Configure the data source with the bind credentials of Directory Proxy Server.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    bind-dn:bind-dn-of-DPS bind-pwd-file:filename
   ```

3. **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

   For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Forwarding Requests as an Alternate User

This section contains information about how to forward requests as an alternate user.

▼ To Configure Remote User Mapping

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Enable operations to be forwarded with an alternate user.**

   ```
   $ dpconf set-server-prop -h host -p port enable-user-mapping:true
   ```

2. **Specify the name of the attribute that contains the ID for remote mapping.**

   ```
   $ dpconf set-server-prop -h host -p port \
    remote-user-mapping-bind-dn-attr:attribute-name
   ```

3. **Enable Directory Proxy Server to map the client ID remotely.**

   ```
   $ dpconf set-server-prop -h host -p port enable-remote-user-mapping:true
   ```

4. **Configure the default mapping.**

   ```
   $ dpconf set-server-prop -h host -p port \
    user-mapping-default-bind-dn:default-mapping-bind-dn \
    user-mapping-default-bind-pwd-file:filename
   ```

   If the mapped identity is not found on the remote LDAP server, the client identity is mapped to the default identity.

5. **Configure the user mapping in the entry for the client on the remote LDAP server.**

   For information about configuring user mapping in Directory Server, see "Proxy Authorization" on page 174.

To Configure Local User Mapping

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Enable operations to be forwarded with an alternate user.**

   ```
   $ dpconf set-server-prop -h host -p port enable-user-mapping:true
   ```

2. **Ensure that Directory Proxy Server is not configured to map the client ID remotely.**

   ```
   $ dpconf set-server-prop -h host -p port enable-remote-user-mapping:false
   ```

3. **Configure the default mapping.**

   ```
   $ dpconf set-server-prop -h host -p port \
    user-mapping-default-bind-dn:default-mapping-bind-dn \
    user-mapping-default-bind-pwd-file:filename
   ```

   The client ID is mapped to this DN if the mapping on the remote LDAP server fails.

4. **If you permit unauthenticated users to perform operations, configure the mapping for unauthenticated clients.**

   ```
   $ dpconf set-server-prop -h host -p port \
    user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
    user-mapping-anonymous-bind-pwd-file:filename
   ```

   For information about how to permit unauthenticated users to perform operations, see "To Configure Anonymous Access" on page 509.

5. **Configure the ID of the client.**

   ```
   $ dpconf set-user-mapping-prop -h host -p port \
    user-bind-dn:client-bind-dn user-bind-pwd-file:filename
   ```

6. **Configure the ID of the alternate user.**

   ```
   $ dpconf set-user-mapping-prop -h host -p port \
    mapped-bind-dn:alt-user-bind-dn mapped-bind-pwd-file:filename
   ```

To Configure User Mapping for Anonymous Clients

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Configure the mapping for unauthenticated clients.**

  ```
  $ dpconf set-server-prop -h host -p port \
   user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
   user-mapping-anonymous-bind-pwd-file:filename
  ```

The mapping for anonymous clients is configured in Directory Proxy Server because the remote LDAP server does not contain an entry for an anonymous client.

For information about permitting unauthenticated users to perform operations, see "To Configure Anonymous Access" on page 509.

◆ ◆ ◆ CHAPTER 25

Connections Between Clients and Directory Proxy Server

For an overview of the connections between clients and Directory Proxy Server, connection handlers, and a description of the criteria and policies used in connection handlers, see Chapter 20, "Connections Between Clients and Directory Proxy Server," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Creating, Configuring, and Deleting Connection Handlers

For information about how to create, configure, and delete connection handlers, and to configure affinity for data views, see the following procedures.

▼ To Create a Connection Handler

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Create a connection handler.

```
$ dpconf create-connection-handler -h host -p port connection-handler-name
```

2 (Optional) View the list of connection handlers.

```
$ dpconf list-connection-handlers -h host -p port
```

▼ To Configure a Connection Handler

Before You Begin The properties of a connection handler must be defined in relation to the properties of the other connection handlers that are defined for the Directory Proxy Server instance. Consider the properties of all of your connection handlers to ensure that they specify different sets of criteria and are prioritized correctly.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View a verbose list of connection handlers to see their key properties and relative priorities.

```
$ dpconf list-connection-handlers -h host -p port -v
Name                              is-enabled  priority  description
--------------------------------  ----------  --------  --------------------------
anonymous                         false       99        unauthenticated connections
default connection handler        true        100       default connection handler
directory services administrators  true        1         Administrators connection handler
```

The connection handlers `anonymous` and `default connection handler` are created when you create an instance of Directory Proxy Server.

2 View all of the properties of one connection handler.

```
$ dpconf get-connection-handler-prop -h host -p port connection-handler-name
```

The default properties of a new connection handler are as follows:

```
aci-source                              :  none
allowed-auth-methods                    :  anonymous
allowed-auth-methods                    :  sasl
allowed-auth-methods                    :  simple
allowed-ldap-ports                      :  ldap
allowed-ldap-ports                      :  ldaps
bind-dn-filters                         :  any
close-client-connection                 :  false
data-view-routing-custom-list           :  none
data-view-routing-policy                :  all-routable
data-view-use-internal-client-identity  :  false
description                             :  -
domain-name-filters                     :  any
enable-data-view-affinity               :  false
group-dn-filters                        :  any
group-search-bind-dn                    :  any
group-search-bind-pwd                   :  none
ip-address-filters                      :  any
is-enabled                              :  false
is-ssl-mandatory                        :  false
priority                                :  99
```

```
request-filtering-policy                :  no-filtering
require-data-view-availability          :  true
resource-limits-policy                  :  no-limits
schema-check-enabled                    :  false
user-filter                             :  any
```

3 Configure the priority of the connection handler.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name\
 priority:value
```

The priority can be any number from 1 to 100, where 1 is the highest priority. For an instance of Directory Proxy Server, the connection handlers are evaluated in order of priority.

Note – You cannot set the priority of a connection handler to `100` because `100` is already set as the priority of the default connection handler.

---

4 (Optional) Specify the DN filtering property of the connection handler.

This property enables you to control access based on part or all of the bind DN. The value of the property is a regular expression.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 bind-dn-filters:regular-expression
```

The bind DN filter takes the form of a Java™ regular expression. For information about creating Java regular expressions, see `http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html`.

For example, to send all binds from users under `ou=people,dc=example,dc=com` to a connection handler named `secure-handler`, set the `bind-dn-filters` property as follows:

```
$ dpconf set-connection-handler-prop -h host1 -p 1389 secure-handler \
 bind-dn-filters:"uid=.*,ou=people,dc=example,dc=com"
```

5 (Optional) Specify the name of a request filtering policy to use with this connection handler.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

where *policy-name* is the name of an existing request filtering policy. For information about how to create and configure a request filtering policy, see "Creating and Configuring Request Filtering Policies and Search Data Hiding Rules" on page 497.

6 (Optional) Specify the name of a resource limits policy to use with this connection handler.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

where *policy-name* is the name of an existing resource limits policy. For information about how to create and configure a resource limits policy, see "Creating and Configuring a Resource Limits Policy" on page 501.

7 Configure any other properties that are listed in Step 2.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 property:value [property:value ...]
```

For example, configure the connection handler to accept SSL connections only.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 is-ssl-mandatory:true
```

For a description of a property and a list of its valid values, run this command:

```
$ dpconf help-properties connection-handler
```

Configure `group-dn-filters`, `group-search-bind-dn`, `group-search-bind-pwd`, and `group-search-bind-pwd-file` to specify the criteria to select connection handlers. For more information, see the respective man pages.

8 Enable the connection handler.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name\
 is-enabled:true
```

9 If necessary, restart the instance of Directory Proxy Server for the changes to take effect.

For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

▼ To Delete a Connection Handler

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 (Optional) View the list of connection handlers.

```
$ dpconf list-connection-handlers -h host -p port
```

2 Delete one or more connection handlers.

```
$ dpconf delete-connection-handler -h host -p port connection-handler-name\
 [connection-handler-name ... ]
```

▼ To Configure Affinity for Data Views

When a connection is allocated to a connection handler, you can use affinity to expose the requests on that connection to the list of data views that are configured for that connection handler, or to all of the configured data views. Therefore, successive requests on that connection are exposed exclusively to the data view that is used for the first request.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Enable affinity for data views.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 enable-data-view-affinity:true
```

2 (Optional) Configure the connection handler to route requests to a custom list of data views.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name\
 data-view-routing-policy:custom
```

3 (Optional) Configure the list of data views.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list:view-name [data-view-routing-custom-list:view-name ...]
```

To add a data view to an existing list of data views, use this command:

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list+:view-name
```

To remove a data view from an existing list of data views, use this command:

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list-:view-name
```

Creating and Configuring Request Filtering Policies and Search Data Hiding Rules

For an overview of request filtering policies, see "Request Filtering Policies for Connection Handlers" in *Sun Directory Server Enterprise Edition 7.0 Reference*. For an overview of search data hiding rules, see "Search Data Hiding Rules in the Request Filtering Policy" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

For information about how to create and configure request filtering policies and search data hiding rules, see the following procedures.

▼ To Create a Request Filtering Policy

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Create a request filtering policy.

```
$ dpconf create-request-filtering-policy policy-name
```

2 Associate the request filtering policy with a connection handler.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

▼ To Configure a Request Filtering Policy

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the properties of a request filtering policy.

```
$ dpconf get-request-filtering-policy-prop -h host -p port policy-name
```

The default properties of a request filtering policy are as follows:

```
allow-add-operations                :  true
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  true
allow-extended-operations           :  true
allow-inequality-search-operations  :  true
allow-modify-operations             :  true
allow-rename-operations             :  true
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
allowed-search-scopes               :  one-level
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ""
description                         :  -
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

2 Configure the request filtering policy by setting one ore more of the properties listed in Step 1.

```
$ dpconf set-request-filtering-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

By setting the properties listed in Step 1, you configure the following features of the request filtering policy:

- The types of operations that clients are allowed to perform
- The subtrees that are exposed to a client or hidden from a client
- The scope for search operations
- The types of search filters
- The attribute types that can or cannot be compared in search and compare operations

▼ To Create Search Data Hiding Rules

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Create one or more search data hiding rules for a request filtering policy.

```
$ dpconf create-search-data-hiding-rule -h host -p port policy-name rule-name \
 [rule-name ...]
```

2 View the properties of a search data hiding rule.

```
$ dpconf get-search-data-hiding-rule-prop policy-name rule-name
```

The default properties of a search data hiding rule are as follows:

```
attrs                           :  none
rule-action                     :  hide-entry
target-attr-value-assertions    :  none
target-dn-regular-expressions   :  none
target-dns                      :  none
```

3 Configure a search data hiding rule by setting one or more of the properties listed in Step 2.

```
$ dpconf set-search-data-hiding-rule-prop -h host -p port policy-name rule-name \
  property:value [property:value ...]
```

One of the following rule actions can be used:

`hide-entry`	The target entry is not returned.
`hide-attributes`	The target entry is returned but the specified attributes are filtered out.
`show-attributes`	The target entry is returned but the unspecified attributes are filtered out.

The rule can be applied to the following entries:

`target-dns`	Entries with the specified DN
`target-dn-regular-expressions`	Entries with the specified DN pattern

`target-attr-value-assertions`	Entries with a specified attribute name and attribute value pair (*attrName#attrValue*)

The following configuration defines a search data hiding rule that hides entries of type `inetorgperson`.

```
$ dpconf set-search-data-hiding-rule-prop -h host1 -p port my-policy my-rule \
  target-attr-value-assertions:objectclass#inetorgperson
```

Example Request Filtering Policy and Search Data Hiding Rule

The following examples contain a request filtering policy and a search data hiding rule. When the request filtering policy is combined with the search data hiding rule, access to data is limited as follows:

- The following types of operations are disallowed: add, delete, extended, modify, and rename.
- Only the `ou=people,dc=sun,dc=com` subtree can be accessed.
- Entries other than `inetorgperson` type are returned by search operations.

EXAMPLE 25–1 Sample Request Filtering Policy

```
allow-add-operations                :  false
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  false
allow-extended-operations           :  false
allow-inequality-search-operations  :  true
allow-modify-operations             :  false
allow-rename-operations             :  false
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
allowed-search-scopes               :  one-level
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ou=people,dc=sun,dc=com
description                         :  myRequestFilteringPolicy
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

EXAMPLE 25–2 Sample Search Data Hiding Rule

```
attrs                              :  -
rule-action                        :  hide-entry
target-attr-value-assertions       :  objectclass:inetorgperson
target-dn-regular-expressions      :  -
target-dns                         :  -
```

Creating and Configuring a Resource Limits Policy

For an overview of resource limits policies, see "Resource Limits Policies for Connection Handlers" in *Sun Directory Server Enterprise Edition 7.0 Reference*. For information about how to create and configure resource limits policies and to customize search limits, see the following procedures.

▼ To Create a Resource Limits Policy

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 Create a resource limits policy.

```
$ dpconf create-resource-limits-policy -h host -p port policy-name
```

For information about how to modify the properties of a resource limits policy, see "To Configure a Resource Limits Policy" on page 501.

2 Associate the resource limits policy to a connection handler.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

▼ To Configure a Resource Limits Policy

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the properties of a resource limits policy.

```
$ dpconf get-resource-limits-policy-prop -h host -p port policy-name
```

The default properties of a resource limits policy are as follows:

```
denied-presence-filter-attr                 :  all
denied-presence-filter-attr-enabled         :  false
description                                 :  -
max-client-connections                      :  unlimited
max-connections                             :  unlimited
max-op-count-per-interval                   :  inlimited
max-simultaneous-operations-per-connection  :  unlimited
max-total-operations-per-connection         :  unlimited
minimum-search-filter-substring-length      :  unlimited
op-count-per-interval-timeout               :  1s
referral-bind-policy                        :  default
referral-hop-limit                          :  default
referral-policy                             :  default
search-size-limit                           :  unlimited
search-time-limit                           :  unlimited
warning-op-count-per-interval               :  unlimited
```

2 **Configure the resource limits policy by setting one or more of the properties that are listed in Step 1:**

```
$ dpconf set-resource-limits-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

To specify the threshold number of operations per time interval at which a warning is raised, run the following command:

```
$ dpconf  set-resource-limits-policy-prop -h host -p port policy-name \
warning-op-count-per-interval:1500
```

When the specified number of operations exceed in a specified time interval, the warning-resource-limit-exceeded alert is raised. For more information on warning-resource-limit-exceeded, see "Configuring Administrative Alerts for Directory Proxy Server" on page 530.

▼ To Block Presence Filters in the Search Operation

1 **Configure `denied-presence-filter-attr` to deny access when search operation contains at least one of the attributes in the list of denied filter attributes.**

```
$ dpconf set-resource-limits-policy-prop  -h host -p port policy-name \
denied-presence-filter-attr:attribute-name
```

2 **Turn on `denied-presence-filter-enabled` to indicate whether to deny access when the search filter contains specified attributes.**

```
$ dpconf set-resource-limits-policy-prop -h host -p port policy-name\
denied-presence-filter-enabled:on
```

▼ To Customize Search Limits

Customized limits can be defined for search operations according to the search base and search scope. If the target DN and scope of a search operation matches the specified criteria, the maximum size of the search result is limited.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Create one or more custom search limits.**

```
$ dpconf create-custom-search-size-limit -h host -p port policy-name \
  custom-search-limit-name [custom-search-limit-name ...]
```

2 **Set the criteria for the custom search limit.**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name one-level-search-base-dn:value subtree-search-base-dn:value
```

3 **Set the limit for the number of results that are returned when a search meets one of the criteria in Step 2.**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
 custom-search-limit-name search-size-limit:value
```

4 **View the properties of a custom search limit.**

```
$ dpconf get-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name
```

The default properties of a custom search limit are as follows:

```
one-level-search-base-dn  :  none
search-size-limit         :  unlimited
subtree-search-base-dn    :  none
```

▼ To Limit LDAP Operations Rates

Directory Proxy Server lets you set a threshold for the maximum number of LDAP operations allowed in a given time period. You set the operations rate limit per connection handler using a resource limits policy. The settings effectively allow you to limit the LDAP operation rate for an

LDAP client application. For example you can use this capability to ensure that one LDAP client application can perform a maximum of 2500 LDAP operations per second, whereas another LDAP client operation is limited to a maximum of 1200 operations per second.

First set up a connection handler to describe connections from the client application whose LDAP operation rate you want to limit. Then create a resource limits policy for the connection handler. Finally follow the steps here to limit the operation rate using the resource limits policy on the connection handler.

1 **Enable the operations rate limit counters.**

```
$ dpconf set-resource-limits-policy-prop -h host -p port policy-name \
max-op-count-per-interval:2500
$ dpconf set-resource-limits-policy-prop -h host -p port policy-name \
op-count-per-interval-timeout:1s
```

2 **When an LDAP client exceeds the operation rate limit you set, Directory Proxy Server can raise an alert provided you set up Directory Proxy Server as described in the "Configuring Administrative Alerts for Directory Proxy Server" on page 530 section.**

To add an alert about operation rate limits being reached, run this command:

```
$ dpconf set-server-prop -h host -p port\
 enabled-admin-alerts+:error-resource-limit-exceeded
```

Directory Proxy Server raises an alert when the operations rate limit is exceeded. Directory Proxy Server also writes a message in the access log each time an operation is refused because the application exceeds its limit.

Configuring Directory Proxy Server as a Connection Based Router

Directory Proxy Server 5.2 is a connection based router. In Directory Proxy Server 5.2, a client connection is routed to a specific directory server. All requests on that client connection are sent to the same directory server until the connection is broken or until the client unbinds.

Directory Proxy Server 7.0 is an operation based router. However, for compatibility, this version of Directory Proxy Server can be configured as a connection based router, as described in the following procedure.

▼ To Configure Directory Proxy Server as a Connection Based Router

1. **Create and configure one or more connection handlers as described in "Creating, Configuring, and Deleting Connection Handlers" on page 493.**

   You can also use the default connection handler.

2. **Configure all connection handlers to route requests to the `root data view` only.**

   For example:

   ```
   $ dpconf set-connection-handler-prop -h host1 -p 1389 myConnectionHandler \
    data-view-routing-policy:custom data-view-routing-custom-list:"root data view"
   ```

3. **Create and configure a data source for each back-end LDAP server as described in "Creating and Configuring LDAP Data Sources" on page 373.**

   For example:

   ```
   $ dpconf create-ldap-data-source -h host1 -p 1389 myDataSource host2:2389
   ```

4. **Create and configure a data source pool as described in "Creating and Configuring LDAP Data Source Pools" on page 376.**

   For example:

   ```
   $ dpconf create-ldap-data-source-pool -h host1 -p 1389 myDataSourcePool
   ```

5. **Attach all of the data sources to the data source pool as described in "Attaching LDAP Data Sources to a Data Source Pool" on page 377.**

   For example,

   ```
   $ dpconf attach-ldap-data-source -h host1 -p 1389 myDataSourcePool myDataSource
   ```

6. **Configure each data source to authenticate clients by using BIND replay as described in "Forwarding Requests With Bind Replay" on page 487.**

   For example:

   ```
   $ dpconf set-ldap-data-source-prop -h host1 -p 1389 myDataSource \
    client-cred-mode:use-client-identity
   ```

7. **Configure affinity between the client connection and the data source pool as described in "Configuring Client Affinity" on page 414.**

   For example:

   ```
   $ dpconf set-ldap-data-source-pool-prop -h host1 -p 1389 myDataSourcePool \
    enable-client-affinity:true client-affinity-policy:read-write-affinity-after-write
   ```

◆ ◆ ◆ CHAPTER 26

Directory Proxy Server Client Authentication

For an overview of client authentication in Directory Proxy Server, see Chapter 21, "Directory Proxy Server Client Authentication," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Configuring Listeners Between Clients and Directory Proxy Server

Directory Proxy Server provides a secure listener and a non-secure listener for communication with clients. For information about listeners for Directory Proxy Server, see "Directory Proxy Server Client Listeners" in *Sun Directory Server Enterprise Edition 7.0 Reference*. This section describes how to configure the listeners.

▼ To Configure the Listeners Between a Client and Directory Proxy Server

Note – This procedure configures the non-secure listener between a client and Directory Proxy Server. To configure the secure listener, perform the same procedure but replace `ldap` with `ldaps`.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help. In DSCC, you can configure this property on the Performance tab.

1 **View the properties of the non-secure listener.**

```
$ dpconf get-ldap-listener-prop -h host -p port
```

The default properties of the non-secure listener are as follows:

```
connection-idle-timeout          :  1h
connection-read-data-timeout     :  2s
connection-write-data-timeout    :  1h
is-enabled                       :  true
listen-address                   :  0.0.0.0
listen-port                      :  port-number
max-connection-queue-size        :  128
max-ldap-message-size            :  unlimited
number-of-threads                :  2
use-tcp-keep-alive               :  true
use-tcp-no-delay                 :  true
```

2 **Change one or more of properties that are listed in Step 1 according to your requirements.**

```
$ dpconf set-ldap-listener-prop -h host -p port property:new-value
```

For example, to disable the non-secure port for an instance of Directory Proxy Server running on host1, run the following command:

```
$ dpconf set-ldap-listener-prop -h host1 -p 1389 is-enabled:false
```

Caution – If you plan to use a privileged port number, you must run Directory Proxy Server as root.

To change the non-secure port number, run the following command:

```
$ dpconf set-ldap-listener-prop -h host -p port listen-port:new-port-number
```

3 **If necessary, restart the instance of Directory Proxy Server for the changes to take effect.**

Changes to certain listener properties require a server restart. `dpconf` alerts you if the server must be restarted. For information about restarting Directory Proxy Server, see "To Restart Directory Proxy Server" on page 364.

Authenticating Clients to Directory Proxy Server

By default, Directory Proxy Server is configured for simple bind authentication. No additional configuration is required for simple bind authentication.

For information about authentication between clients and Directory Proxy Server, see "Client Authentication Overview" in *Sun Directory Server Enterprise Edition 7.0 Reference*. For information about how to configure authentication, see the following procedures.

To Configure Certificate-based Authentication

For information about certificate-based authentication of clients, see "Configuring Certificates in Directory Proxy Server" in *Sun Directory Server Enterprise Edition 7.0 Reference*. This section describes how to configure certificate-based authentication.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

Note – Certificate-based authentication can only be performed over an SSL connection.

- **Configure Directory Proxy Server to require a client to present a certificate when the client establishes an SSL connection.**

```
$ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
```

To Configure Anonymous Access

For information about anonymous access, see "Anonymous Access" in *Sun Directory Server Enterprise Edition 7.0 Reference*. For information about how to map the identity of an anonymous client to another identity, see "Forwarding Requests as an Alternate User" on page 490.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Permit unauthenticated users to perform operations.**

```
$ dpconf set-server-prop -h host -p port \
allow-unauthenticated-operations:true
```

2. **(Optional) Specify the access mode for unauthenticated users.**

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations-mode:mode
```

For more information, see `allow-unauthenticated-operations-mode(5dpconf)`.

To Configure Directory Proxy Server for SASL External Bind

For information about SASL external bind, see "Using SASL External Bind" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Disallow unauthenticated operations.**

   ```
   $ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
   ```

2. **Require clients to present a certificate when establishing a connection.**

   ```
   $ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
   ```

   The client provides a certificate that contains a DN.

3. **Enable the authentication of clients by SASL external bind.**

   ```
   $ dpconf set-server-prop -h host -p port -e allow-sasl-external-authentication:true
   ```

4. **Configure the identity used by Directory Proxy Server to map a client certificate on a back-end LDAP server.**

   ```
   $ dpconf set-server-prop -h host -p port -e \
   cert-search-bind-dn:bind-DN cert-search-bind-pwd-file:filename
   ```

5. **Configure the base DN of the subtree that Directory Proxy Server searches.**

   Directory Proxy Server searches the subtree to find a user entry that is mapped to a client certificate.

   ```
   $ dpconf set-server-prop -h host -p port -e  \
   cert-search-base-dn:base-DN
   ```

6. **Map information in the client certificate to certificates on the LDAP server.**

   a. **Name the attribute on the LDAP server that contains certificates.**

      ```
      $ dpconf set-server-prop -e cert-search-user-attribute:attribute
      ```

   b. **Map an attribute on the client certificate to the DN of the entry on the LDAP server that contains certificates.**

      ```
      $ dpconf set-server-prop -h host -p port -e \
       cert-search-attr-mappings:client-side-attribute-name:server-side-attribute-name
      ```

      For example, to map a client certificate with the DN `cn=user1,o=sun,c=us` to an LDAP entry with the DN `uid=user1,o=sun`, run the following command:

      ```
      $ dpconf set-server-prop -h host1 -p 1389 -e cert-search-attr-mappings:cn:uid \
       cert-search-attr-mappings:o:o
      ```

7. **(Optional) Route requests for SASL external bind operations to all data views or to a custom list of data views.**

   - To route requests to all data views, run this command:

     ```
     $ dpconf set-server-prop -h host -p port -e \
     cert-data-view-routing-policy:all-routable
     ```

- To route requests to a list of data views, run this command:

```
$ dpconf set-server-prop -h host -p port -e cert-data-view-routing-policy:custom \
cert-data-view-routing-custom-list:view-name [view-name...]
```

Troubleshooting Use the `-e` option wherever it is mentioned in the above procedure to successfully configure Directory Proxy Server for SASL External Bind.

◆ ◆ ◆ CHAPTER 27

Directory Proxy Server Logging

Directory Proxy Server logs information in access logs and error logs. Unlike Directory Server, Directory Proxy Server does not have an audit log. For a description of the logs in Directory Proxy Server, see Chapter 23, "Directory Proxy Server Logging," in *Sun Directory Server Enterprise Edition 7.0 Reference*.

This chapter covers the following topics:

Viewing Directory Proxy Server Logs

You can view Directory Proxy Server logs directly through the log files or by using Directory Service Control Center (DSCC).

By default, the logs are stored in this directory:

instance-path`/logs`

The following figure shows a screen capture of the error log for Directory Proxy Server on DSCC.

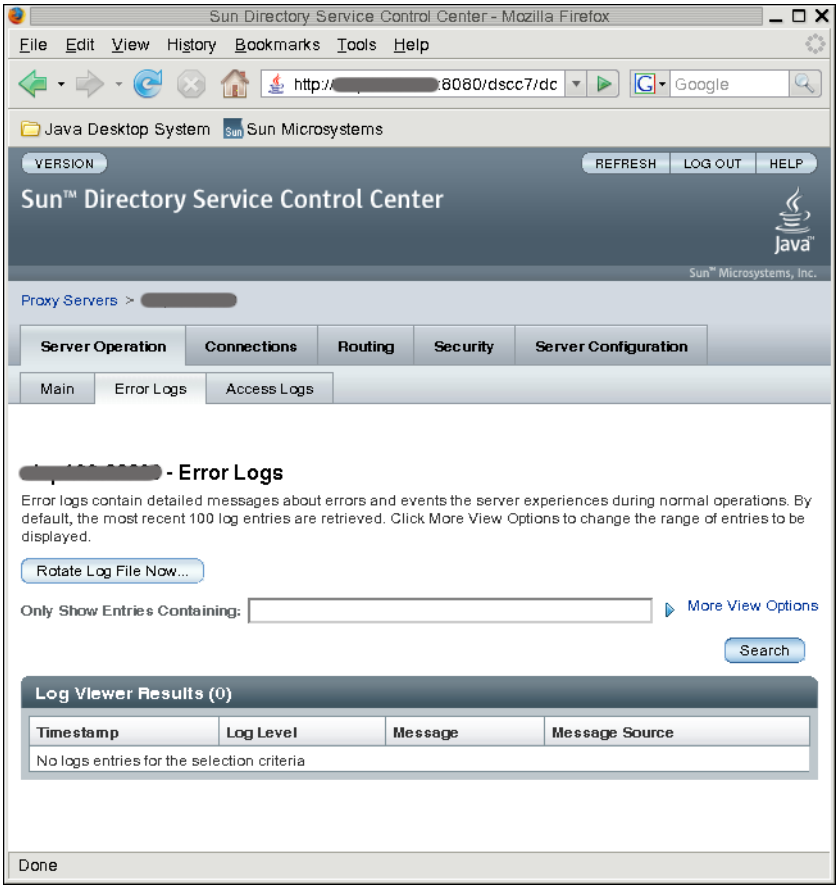


FIGURE 27–1 Error Log Window for Directory Proxy Server

Configuring Directory Proxy Server Logs

Directory Proxy Server error logs and access logs can be configured by using the `dpconf` command or DSCC. For information about how to configure the logs by using DSCC, see the Directory Proxy Server online help. This section describes how to configure Directory Proxy Server logs by using the `dpconf` command.

You can retrieve a complete list of the configuration options along with the allowed values and default values by running these commands:

```
$ dpconf help-properties error-log
$ dpconf help-properties access-log
```

▼ To Configure Directory Proxy Server Access and Error Logs

This procedure configures the Directory Proxy Server access log. To configure the Directory Proxy Server error log, perform the same procedure but replace `access` with `error`.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 View the properties of the access log.

```
$ dpconf get-access-log-prop -h host -p port
```

The default properties of an access log are as follows:

```
default-log-level                :  info
enable-log-rotation              :  true
log-buffer-size                  :  1M
log-file-name                    :  logs/access
log-file-perm                    :  600
log-level-client-connections     :  inherited
log-level-client-disconnections  :  inherited
log-level-client-operations      :  inherited
log-level-connection-handlers    :  inherited
log-level-data-sources           :  inherited
log-level-data-sources-detailed  :  none
log-min-size                     :  100M
log-rotation-frequency           :  1h
log-rotation-policy              :  size
log-rotation-size                :  100M
log-rotation-start-day           :  -
log-rotation-start-time          :  -
log-search-filters               :  false
max-age                          :  unlimited
max-log-files                    :  10
max-size                         :  unlimited
min-free-disk-space-size         :  1M
```

2 Change one or more of the properties that are listed in Step 1.

```
$ dpconf set-access-log-prop -h host -p port property:value \
 [property:value ...]
```

For example, to set the default log level for all message categories to `all`, set the value of the `default-log-level` property to `all`.

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:all
```

To disable all logs, irrespective of the log level for each message category, set the value of the `default-log-level` property to `none`.

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:none
```

To reset a specific log level to the default log level, set that log level property to `inherited`. For example, to reset the log level for client connections, run the following command:

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-level-client-connections:inherited
```

For information about properties that can be set by the `set-access-log-prop` subcommand, type:

```
$ dpconf help-properties access-log
```

Configuring Directory Proxy Server Log Rotation

By default, log files are rotated when the log file size reaches 100 Mbytes. Ten log files are retained by default, after which the rotation procedure begins to overwrite the oldest log file. This section describes how to configure Directory Proxy Server logs for scheduled rotation, how to rotate logs manually, and how to disable log rotation. For example configurations, see "Example Configurations for Log Rotation" on page 518.

▼ To Configure Periodic Rotation of Access and Error Logs

This procedure configures the Directory Proxy Server access log. To configure the Directory Proxy Server error log, perform the same procedure but replace `access` with `error`.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **(Optional) View the properties of the access log.**

   ```
   $ dpconf get-access-log-prop -h host -p port
   ```

2. **(Optional) View valid values for the properties of the access log.**

   ```
   $ dpconf help-properties access-log
   ```

3. **To rotate logs when they reach a certain size, set the following properties:**

   ```
   $ dpconf set-access-log-prop -h host -p port \
    log-rotation-policy:size log-rotation-size:maximum file size
   ```

Caution – In case of high activity levels, because of the asynchronous nature of Directory Proxy Server, the log file might not be rotated at the exact configured size but at a size close to the configured size. This means that the rotated file might end up being slightly smaller or slightly larger than the configured size.

If the unit of the maximum file size is not specified, the default unit of *bytes* is used. When the log file reaches the defined size, the log is rotated. The file size must be at least 1 Mbyte and no more than 2 Gbytes.

For an example of how to rotate logs by size, see "Rotating the Log Based on Log Size" on page 518.

4 **To rotate logs periodically, irrespective of the log size, set the following properties:**

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-frequency:interval in months, weeks, hours, or minutes \
 log-rotation-policy:periodic \
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

If the log is configured for rotation on the 31st of the month but the month has fewer than 31 days, the log is rotated on the first day of the following month.

By default, the `log-rotation-start-day` and `log-rotation-start-time` properties have no default value. If you configure to rotate logs without setting these properties, the log will be rotated as per the specified frequency but the time of the day or day of the week might be changed.

For examples of how to rotate logs periodically, see "Rotating the Log Based on Time" on page 518.

5 **To rotate logs periodically if the log file is big enough, set the `log-rotation-frequency` and `log-min-size` properties.**

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-frequency:interval in months, weeks, hours, or minutes \
 log-rotation-policy:periodic log-min-size:minimum file size
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

The `log-min-size` property represents the minimum size of the log. The rotation takes place at the scheduled time only if the log file is bigger than the specified size.

If the log is configured for rotation on the 31st of the month but the month has fewer than 31 days, the log is rotated on the first day of the following month.

By default, the `log-rotation-start-day` and `log-rotation-start-time` properties have no default value. If you configure to rotate logs without setting these properties, the log will be rotated as per the specified frequency but the time of the day or day of the week might be changed.

For an example of how to rotate logs periodically if the file size is big enough, see "Rotating the Log Based on Time and Log Size" on page 519.

▼ To Rotate Access and Error Logs Files Manually

This procedure rotates the Directory Proxy Server access log. To rotate the Directory Proxy Server error log, perform the same procedure but replace `access` with `error`.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Rotate the access log.**

```
$ dpconf rotate-log-now -h host -p port access
```

▼ To Disable Access and Error Log Rotation

This procedure disables rotation of the Directory Proxy Server access log. To disable rotation of the Directory Proxy Server error log, perform the same procedure but replace `access` with `error`.

- **Disable log file rotation.**

```
$ dpconf set-access-log-prop -h host -p port enable-log-rotation:false
```

Example Configurations for Log Rotation

Examples of how to configure log rotation by log size, time, or both follow.

Rotating the Log Based on Log Size

This section example shows how to configure a log rotation according to log size only. This configuration rotates the log when it reaches 10 Mbytes, irrespective of the time since the log was last rotated.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-policy:size \
  log-rotation-size:10M
```

Rotating the Log Based on Time

The examples in this section show how to configure log rotation according to the time since the last rotation, irrespective of log size.

- This configuration rotates the log after every 10 hours, irrespective of log size.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:10h \
 log-rotation-policy:periodic
```

For example, if the log is rotated at 3:00 today, the next rotations will take place after every 10 hours such as 13:00, 23:00, and 9:00 next day. Without setting `log-rotation-start-day` and `log-rotation-start-time`, the rotation might not take place everyday at the same time.

- This configuration rotates the log at 3:00, 13:00, and 23:00 every day, irrespective of the size of the log file. Because the `log-rotation-start-time` parameter takes precedence over the `log-rotation-frequency` parameter, the log is rotated 3:00, that is, 4 hours after the last rotation.

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:10h \
   log-rotation-policy:periodic log-rotation-start-time:0300
  ```

- This configuration rotates the log at noon on Monday, and then at the same time every week, irrespective of the size of the log file.

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1w \
   log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
  ```

- This configuration rotates the log at noon on Monday, and then every 3 days, irrespective of the size of the log file.

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:3d \
   log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
  ```

  The log is rotated on the following days: Monday, Thursday, Sunday, Wednesday, and so on. Notice that the `log-rotation-start-day` parameter applies to the first week only. The log is not rotated on the Monday of the second week.

- This configuration rotates the log at noon on the 22nd day of the month, and then at the same time every month, irrespective of log size.

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1m \
   log-rotation-policy:periodic log-rotation-start-day:22 \
   log-rotation-start-time:1200
  ```

If the `log-rotation-start-day` is set to 31 and the month has only 30 days, the log is rotated on the first day of the following month. If the `log-rotation-start-day` is set to 31 and the month has only 28 days (February), the log is rotated on the 3rd.

Rotating the Log Based on Time and Log Size

This example shows how to configure a log rotation for a specified interval if the file size is big enough.

This configuration rotates the log at 3:00, 11:00, and 19:00 every day, if the size of the log file exceeds 1 Mbyte. If the size of the log file does not exceed 1 Mbyte, the log file is not rotated.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:8h \
 log-rotation-policy:periodic log-min-size:1M log-rotation-start-time:0300
```

Deleting Directory Proxy Server Logs

Directory Proxy Server enables you to configure log deletion based on time, size, or free disk space (the default). For more information about these deletion policies, see "Log File Deletion" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

The following procedures configure log deletion for the access log. To configure log deletion for the error log, use the same commands, but replace `access` with `error`.

▼ To Configure Access and Error Log Deletion Based on Time

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Specify the maximum age for log files.**

  `$ dpconf set-access-log-prop -h` *host* `-p` *port* `max-age:`*duration*

  where *duration* includes a unit of days (`d`), weeks (`w`), or months (`M`). For example, to delete backup log files older than five days, use this command:

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 max-age:5d
  ```

▼ To Configure Access and Error Log Deletion Based on File Size

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Specify the maximum size for log files.**

  `$ dpconf set-access-log-prop -h` *host* `-p` *port* `max-size:`*memory-size*

  For example, to keep only the most recent log files with their aggregate size not more than 5 Mbytes, use this command:

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 max-size:5M
  ```

▼ To Configure Access and Error Log Deletion Based on Free Disk Space

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

- **Specify the minimum available disk space.**

```
$ dpconf set-access-log-prop -h host -p port min-free-disk-space-size:memory-size
```

For example, to delete backup log files when the available disk space is less than 2 Mbytes, use this command:

```
$ dpconf set-access-log-prop -h host1 -p 1389 min-free-disk-space-size:2M
```

Logging Alerts to the syslogd Daemon

This section describes how to configure the logging of alert messages to the syslogd daemon and how to configure the operating system to accept syslog alerts.

▼ To Configure Directory Proxy Server to Log Alerts to the syslogd Daemon

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **(Optional) View the current values of the properties for the system log alerts.**

```
$ dpconf get-server-prop -h host -p port syslog-alerts-enabled \
 syslog-alerts-facility syslog-alerts-host
```

The default properties for the system log alerts are as follows:

```
syslog-alerts-enabled   :  false
syslog-alerts-facility  :  USER
syslog-alerts-host      :  localhost
```

The syslog-alerts-host property defines the host name of the syslogd daemon to which the messages are sent. The syslog-alerts-facility property is read-only and causes messages to be sent to the user category in the system log.

2. **Enable alert messages to be logged to the syslogd daemon.**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
```

3 **(Optional) Send alert messages to the `syslogd` daemon on a different host.**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-host:hostname
```

Configuring the Operating System to Accept `syslog` Alerts

This section provides instructions on configuring the Solaris™, Linux, and HP-UX operating systems to accept `syslog` alerts.

▼ To Configure the Solaris OS to Accept `syslog` alerts

1 **Add the appropriate facility to the `syslog` configuration file.**

For example, to store all alerts using the `USER` facility, add the following line to `/etc/syslog.conf`:

```
user.info       /var/adm/info
```

Here `/var/adm/info` is an example local directory in which messages will be stored. Ensure that `/var/adm/info` exists before continuing.

2 **Restart the `syslogd` daemon.**

a. **On Solaris 8 and 9, restart `syslogd` by typing this:**

```
$ /etc/init.d/syslog stop | start
```

b. **On Solaris 10, restart `syslogd` by typing this:**

```
$ svcadm restart system/system-log
```

3 **Verify that messages are logged in `syslog`.**

```
$ logger -p user.info "Test message"
$ cat /var/adm/info
 Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
```

▼ To Configure Linux to Accept `syslog` Alerts

1 **Add the appropriate facility to the `syslog` configuration file.**

For example, to store all alerts using the `USER` facility, add the following line to `/etc/syslog.conf`:

```
user.info       /var/adm/info
```

Here `/var/adm/info` is an example local directory in which messages will be stored. Ensure that `/var/adm/info` exists before continuing.

2. **Configure the `syslogd` daemon to run with the `-r` option.**

   This option allows `syslogd` to accept connections from the network. By default, the `-r` option is not set.

   To set the `-r` option, add the following line to `/etc/sysconfig/syslog`:

   ```
   SYSLOGD_OPTIONS="-m 0 -r"
   ```

   If `/etc/sysconfig/syslog` does not exist, add the same line to `/etc/init.d/syslog`.

3. **Restart the `syslogd` daemon.**

   ```
   $ /etc/init.d/syslog stop | start
   ```

4. **Verify that messages are logged in `syslog`.**

   ```
   $ logger -p user.info "Test message"
   $ cat /var/adm/info
    Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
   ```

▼ To Configure HP-UX to Accept `syslog` alerts

1. **Add the appropriate facility to the `syslog` configuration file.**

   For example, to store all alerts using the `USER` facility, add the following line to `/etc/syslog.conf`:

   ```
   user.info       /var/adm/info
   ```

   Here `/var/adm/info` is an example local directory in which messages will be stored. Ensure that `/var/adm/info` exists before continuing.

2. **Restart the `syslogd` daemon.**

   ```
   $ /sbin/init.d/syslogd stop | start
   ```

3. **Verify that messages are logged in `syslog`.**

   ```
   $ logger -p user.info "Test message"
   $ cat /var/adm/info
    Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
   ```

Tracking Client Requests Through Directory Proxy Server and Directory Server Access Logs

To track the path of a client request, you must understand how requests are logged in the Directory Proxy Server access log and in the Directory Server access log. To understand this section, first read "Tracking Client Requests Through Directory Proxy Server and Directory Server Access Logs" in *Sun Directory Server Enterprise Edition 7.0 Reference*.

▼ To Track Operations From Directory Server Through Directory Proxy Server to the Client Application

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **Locate the connection number for the operation that you want to track in the Directory Server access log.**

For example, the following line in the access log shows an operation, `op=2` with connection number `conn=12839`.

```
[20/Jul/2006:18:01:49 -0500] conn=12839 op=2 msgId=4 - SRCH base="dc=example,dc=com"
 scope=2 filter="(objectClass=organizationalunit)" attrs=ALL
```

2 **Obtain the Directory Proxy Server connection information for that connection.**

To obtain this information, search the Directory Server access log to locate all operations with the corresponding connection number. For example, on UNIX systems, run the following `grep` command to locate all lines in the Directory Server access log that correspond to connection `conn=12839`:

```
$ grep conn=12839 access
```

The line showing the initial LDAP connection is what you are looking for and will be similar to this:

```
[19/Jul/2006:16:32:51 -0500] conn=12839 op=-1 msgId=-1 - fd=27 slot=27
 LDAP connection from 129.153.160.175:57153 to 129.153.160.175
```

The previous line shows that there is an LDAP connection from **129.153.160.175:57153** to Directory Server. The port number (**57153**) is the information that is required to link the connection back to the Directory Proxy Server access log. The port number enables you to find the corresponding connection in the Directory Proxy Server log, and to locate the client information from this connection.

If the log files have been rotated since the connection was first established, you need to search the archived log files as well as the current access log file.

3 **Locate the corresponding connection in the Directory Proxy Server access log.**

To obtain this information, search the Directory Proxy Server access log to locate all operations with the corresponding port number.

You might find multiple entries in the log file with the same port number. To ensure that you locate the correct entry, include the timestamp from the Directory Server log entry in your search.

For example, on UNIX systems, run the following `grep` command to locate the connection entry that corresponds to the timestamp and port number found in the Directory Server log:

```
$ grep 19/Jul/2006:16:32 access | grep 57153
```

Note that the *seconds* value is excluded from the timestamp to take into account slight differences in server times.

The corresponding line in the Directory Proxy Server log will be similar to this:

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO  - Created BIND  LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153
 server=idm160.central.sun.com:9389 main
```

This line shows that Directory Proxy Server created a `BIND` connection to `s_conn=sunds-d1m1-9389:34`. Directory Proxy Server identifies itself as the client `client=0.0.0.0` on TCP port `57153`.

The important information to extract from this line of the log is the server ID and port number (**s_conn=sunds-d1m1-9389:34**).

4 Locate all operations that correspond to the server ID and port number identified in the previous step.

To obtain this information, search the Directory Proxy Server access log for all operations with the corresponding server ID and port number.

For example, on UNIX systems, run the following `grep` command to locate the operation that corresponds to the server ID found in the previous step:

```
$ grep s_conn=sunds-d1m1-9389:34 access
```

In this case, it is not useful to search for the timestamp because these operations might span several days. However, you must determine that the operations returned by the search are the correct ones. If there are multiple `Create` connection statements, ensure that you locate the one that corresponds to the original search statement. To do this, match the timestamp to the timestamp found in Step 1.

The following extract of the Directory Proxy Server access log shows all operations returned for `s_conn=sunds-d1m1-9389:34`.

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO - Created BIND LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153 server=idm160.central.sun.com:9389 main
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0
BIND dn="cn=directory manager" method="SIMPLE" s_msgid=3 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0
BIND RESPONSE err=0 msg="" s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1
SEARCH base="dc=example,dc=com" scope=2 s_msgid=4 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1
SEARCH RESPONSE err=0 msg="" nentries=1 s_conn=sunds-d1m1-9389:34
```

With this information, you can see that the connection ID for this search operation on Directory Proxy Server is 31 (`conn=31`).

5 Locate the client connection IP address that corresponds to the connection ID found in the previous step.

To obtain this information, search the Directory Proxy Server access log for all operations with the correct connection ID and timestamp. The timestamp to use is the one in the original search statement in Step 1.

For example, on UNIX systems, run the following `grep` command to locate the client connection IP address:

```
$ grep "20/Jul/2006:18:01" access | grep conn=31
```

The line you are interested in is similar to this:

```
[20/Jul/2006:18:01:49 -0500] - CONNECT - INFO  - conn=31 client=129.150.64.156:2031
server=0.0.0.0:11389 protocol=LDAP
```

6 Determine who owns the IP address found in the previous step.

With this information, you can establish precisely who was responsible for the operation performed on Directory Server.

◆ ◆ ◆ CHAPTER 28

Directory Proxy Server Monitoring and Alerts

Monitoring detects failure of Directory Proxy Server and of data sources.

For a description of the monitoring framework for Directory Proxy Server, and for a detailed layout of the `cn=monitor` entry, see "Monitoring Directory Proxy Server" in *Sun Directory Server Enterprise Edition 7.0 Reference*. This chapter covers the following topics:

Retrieving Monitored Data About Directory Proxy Server

To retrieve monitored data about Directory Proxy Server, use the `cn=monitor` entry. This entry is managed by Directory Proxy Server in a local, in-memory database. You can retrieve attributes under `cn=monitor` by performing an LDAP search on the `cn=monitor` entry. You must bind as the Proxy Manager to search this entry.

For information about using the JVM to retrieve monitored data, see "Retrieving Monitored Data About Directory Proxy Server by Using the JVM" on page 533.

Retrieving Monitored Data About Data Sources

For a description of how Directory Proxy Server monitors the health of data sources, see "Monitoring Data Sources" in *Sun Directory Server Enterprise Edition 7.0 Reference*. This section describes how to configure the monitoring of data sources.

Note – In addition to LDAP data source, you can also monitor the health of JDBC data source using `monitoring-inactivity-timeout`, `monitoring-interval`, and `monitoring-mode` properties.

The proactive monitoring is implemented for LDAP data source as well as for JDBC data source. The implementation for both the data sources is not the same as the nature of the data sources is different.

▼ To Monitor a Data Source by Listening for Errors

In this type of monitoring, Directory Proxy Server listens for errors on the traffic between Directory Proxy Server and the data sources. This type of monitoring is called reactive monitoring because Directory Proxy Server reacts if an error is detected, but does not actively test data sources.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Set the monitoring mode for the data source to `reactive`.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:reactive
   ```

2. **Configure an alert to be sent when an error is detected or when a data source goes offline or online, as described in "Configuring Administrative Alerts for Directory Proxy Server" on page 530.**

▼ To Monitor a Data Source by Periodically Establishing Dedicated Connections

Directory Proxy Server creates a dedicated connection to a data source if there have been no requests to or responses from the data source for a specified interval.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Set the monitoring mode for the data source to `proactive`.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
   ```

2. **Configure the monitoring search request that is performed by Directory Proxy Server.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource \
     monitoring-bind-timeout:timeout monitoring-entry-dn:dn \
     monitoring-search-filter:filter monitoring-entry-timeout:timeout
   ```

The following properties are used in the search request:

`monitoring-bind-timeout`	The length of time that Directory Proxy Server waits to establish a connection to the data source. By default, the value of this property is 5 seconds.
`monitoring-entry-dn`	The DN of the target entry in the search request. By default, this property is the root DSE entry ("").
`monitoring-search-filter`	The search filter.
`monitoring-entry-timeout`	The length of time that Directory Proxy Server waits for the search response. By default, the value of this property is 5 seconds.

3 (Optional) Configure the proactive monitoring to bind as a specific user.

```
$ dpconf set-ldap-data-source-prop ldap-data-source \
monitoring-bind-dn:uid=user-id monitoring-bind-pwd-file:password-file
```

Replace the `user-id` with a valid dn such as `uid=bjensen,dc=example,dc=com` and `password-file` with a path to the file containing password.

By default, the bind is performed as anonymous, that is, both the `monitoring-bind-dn` and `monitoring-bind-pwd` attributes are set to `none`.

4 Set the polling interval.

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource \
down-monitoring-interval:interval
```

If a connection is down, Directory Proxy Server polls the connection at this interval to detect its recovery. If the interval is not specified, the value of `monitoring-interval` is used.

5 (Optional) Configure the availability monitor to specify the number of times it will poll the connection when it is first detected as down.

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-retry-count:count
```

6 Configure an alert to be sent when a data source is detected as offline or online, as described in "Configuring Administrative Alerts for Directory Proxy Server" on page 530.

▼ To Monitor a Data Source by Testing Established Connections

In this type of monitoring, Directory Proxy Server performs a search on each connection to each data source at a regular interval. In this way, Directory Proxy Server detects closed connections and prevents connections from being dropped because of inactivity.

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **Set the monitoring mode for the data source to `proactive`.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
   ```

2. **Set the time interval after which Directory Proxy Server sends a request to a data source to prevent connections from being dropped.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource \
    monitoring-inactivity-timeout:time
   ```

   By default, the inactivity timeout is 120 seconds.

3. **(Optional) Configure the proactive monitoring to bind as a specific user.**

   ```
   $ dpconf set-ldap-data-source-prop ldap-data-source
   monitoring-bind-dn:uid=user-id monitoring-bind-pwd-file:password-file
   ```

   Replace the `user-id` with a valid dn such as `uid=bjensen,dc=example,dc=com` and `password-file` with a path to the file containing password.

   By default, the bind is performed as anonymous, that is, both the `monitoring-bind-dn` and `monitoring-bind-pwd` attributes are set to `none`.

4. **Configure an alert to be sent when a data source is detected as offline or online, as described in "Configuring Administrative Alerts for Directory Proxy Server" on page 530.**

Configuring Administrative Alerts for Directory Proxy Server

For information about how to configure administrative alerts, see the following procedures.

▼ To Enable Administrative Alerts

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1. **View the enabled alerts.**

   ```
   % dpconf get-server-prop -h host -p port enabled-admin-alerts
   ```

2. **Enable one or more administrative alerts.**

   ```
   % dpconf set-server-prop -h host -p port enabled-admin-alerts:alert1 \
     [enabled-admin-alerts:alert2 ...]
   ```

For example, to enable all available alerts, run this command:

```
% dpconf set-server-prop -h host -p port \
 enabled-admin-alerts:error-configuration-reload-failure-with-impact \
 enabled-admin-alerts:error-resource-limit-exceeded \
 enabled-admin-alerts:error-server-shutdown-abrupt \
 enabled-admin-alerts:info-configuration-reload \
 enabled-admin-alerts:info-data-source-available \
 enabled-admin-alerts:info-server-shutdown-clean \
 enabled-admin-alerts:info-server-startup \
 enabled-admin-alerts:warning-configuration-reload-failure-no-impact \
 enabled-admin-alerts:warning-data-source-unavailable \
 enabled-admin-alerts:warning-data-sources-inconsistent \
 enabled-admin-alerts:warning-listener-unavailable \
 enabled-admin-alerts:warning-resource-limit-exceeded
```

To disable all email alerts, run this command:

```
% dpconf set-server-prop -h host -p port email-alerts-enabled:false
```

To add an alert to an existing list of enabled alerts, run this command:

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts+:alert-name
```

To remove an alert from an existing list of enabled alerts, run this command:

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts-:alert-name
```

By default, all alerts are enabled. For example, once all the email alerts are enabled (`email-alerts-enabled:true`), run the following command to receive all the email alerts :

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts:all
```

See Also For more information, see `enabled-admin-alerts(5dpconf)`.

▼ To Configure Administrative Alerts to Be Sent to Syslog

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Select the alerts that will be sent to the `syslog` daemon, as described in "To Enable Administrative Alerts" on page 530.**

2 **Enable alerts to be sent to the `syslog` daemon.**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
```

All alerts are sent to the `syslog` with the facility of `USER`.

3 **Set the host name of the `syslog` daemon to which alerts are to be sent.**

```
$ dpconf set-server-prop -h host -p port syslog_hostname:hostname
```

▼ To Configure Administrative Alerts to Be Sent to Email

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Select the alerts that will be sent to the `syslog`, as described in "To Enable Administrative Alerts" on page 530.**

2 **Configure the address and characteristics of the email.**

```
$ dpconf set-server-prop -h host -p port email-alerts-smtp-host:host-name \
  email-alerts-smtp-port:port-number \
  email-alerts-message-from-address:sender-email-address \
  email-alerts-message-to-address:receiver-email-address \
  [email-alerts-message-to-address:receiver-email-address ...] \
  email-alerts-message-subject:email-subject
```

3 **Enable alerts to be sent to email.**

```
$ dpconf set-server-prop -h host -p port email-alerts-enabled:true
```

4 **(Optional) Set a flag to include the alert code in the email**

```
$ dpconf set-server-prop -h host -p port \
 email-alerts-message-subject-includes-alert-code:true
```

▼ To Configure Administrative Alerts to Run a Script

You can use DSCC to perform this task. For information, see "Directory Service Control Center Interface" on page 537 and the DSCC online help.

1 **Select the alerts that will be sent to the `syslog`, as described in "To Enable Administrative Alerts" on page 530.**

2 **Enable alerts to run a script.**

```
$ dpconf set-server-prop -h host -p port scriptable-alerts-enabled:true
```

3 **Set the name of the script that will be run.**

```
$ dpconf set-server-prop -h host -p port scriptable-alerts-command:script-name
```

Retrieving Monitored Data About Directory Proxy Server by Using the JVM

Directory Proxy Server runs inside a Java Virtual Machine (JVM) and depends on the memory of the JVM machine. To ensure that Directory Proxy Server is running correctly, you must monitor the memory consumption of the JVM machine.

For information about how to tune parameters for the JVM machine, see "Hardware Sizing For Directory Proxy Server" in *Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide*.

By default, the heap size of the JVM machine is 250 Mbytes. If Directory Proxy Server does not have enough physical memory, the heap size might be less than 250 Mbytes.

When Directory Proxy Server is running, you can monitor the heap size of the JVM machine to ensure that it is not running out of memory. To do this, use the standard tools delivered with the Java Development Kit (JDK). These tools are located in these directories: `$JAVA_HOME/bin/jps` and `$JAVA_HOME/bin/jstat`.

▼ To View the Heap Size of the JVM

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

- **View the heap size of JVM.**

```
$ dpadm get-flags instance-path jvm-args
jvm-args: -Xms250M  -Xmx250M
```

▼ To Monitor the Heap Size of JVM When Directory Proxy Server is Running

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **View the PID of your instance of Directory Proxy Server.**

```
$ jps
```

2 View the memory used by the JVM machine.

```
$ jstat -gcutil PID
```

- If the zero column is near to 100%, the JVM machine does not have enough memory.
- FGC is the number of full garbage collection (GC) events. Garbage collection is expansive.
- GCT (garbage collection time) is the amount of time spent by the GC.

PART III

Directory Service Control Center Administration

◆ ◆ ◆ CHAPTER 29

Directory Service Control Center Configuration

This chapter includes information related to DSCC interface and DSCC configuration.

Directory Service Control Center Interface

Directory Service Control Center (DSCC) is a user interface that enables you to manage Directory Servers and Directory Proxy Servers by using a browser.

To configure DSCC, see "Configuring DSCC" on page 542. For information about using DSCC, see the following sections.

Administration Users for DSCC

DSCC has a few administration logins.

- **OS user.** Creates a server instance and is the only user who has the right to run operating system commands on a server instance by using the `dsadm` command. DSCC might request the OS user password in some cases. This user must have a password and must be able to create directory server instances.
- **Directory Manager.** The LDAP superuser for a server. The default DN is `cn=Directory Manager`.
- **Directory Administrator.** Administers a Directory Server. This user has the same rights as the Directory Manager but are subject to access controls, password policies, and authentication requirements. You can create as many Directory Administrators as you need.

- **Directory Service Manager**. Manages server configuration and data on multiple machines through DSCC. This user has the same rights as the Directory Manager for each of the servers registered in DSCC and is a member of the Directory Administrators Group.

▼ To Access DSCC

If you experience any difficulty accessing DSCC, see Chapter 8, "Troubleshooting DSCC Problems," in *Sun Directory Server Enterprise Edition 7.0 Troubleshooting Guide*.

1. **Ensure that DSCC has been correctly installed, as described in Chapter 2, "Installing Directory Server Enterprise Edition," in *Sun Directory Server Enterprise Edition 7.0 Installation Guide*.**

2. **Access DSCC directly in your preferred application server by typing the DSCC host URL. DSCC host URL can be any of the following depending on the configuration of your application server.**

   `https://`*hostname*`:8181/dscc7`

   or

   `http://`*hostname*`:8080/dscc7`

   where hostname is the system on which you installed the DSCC software.

3. **Log in to DSCC.**

   If this is the first time that you log in to DSCC, you must set the Directory Service Manager password. On subsequent logins, use the password that you set on the first login.

   You are now logged into DSCC and at the Common Tasks tab.

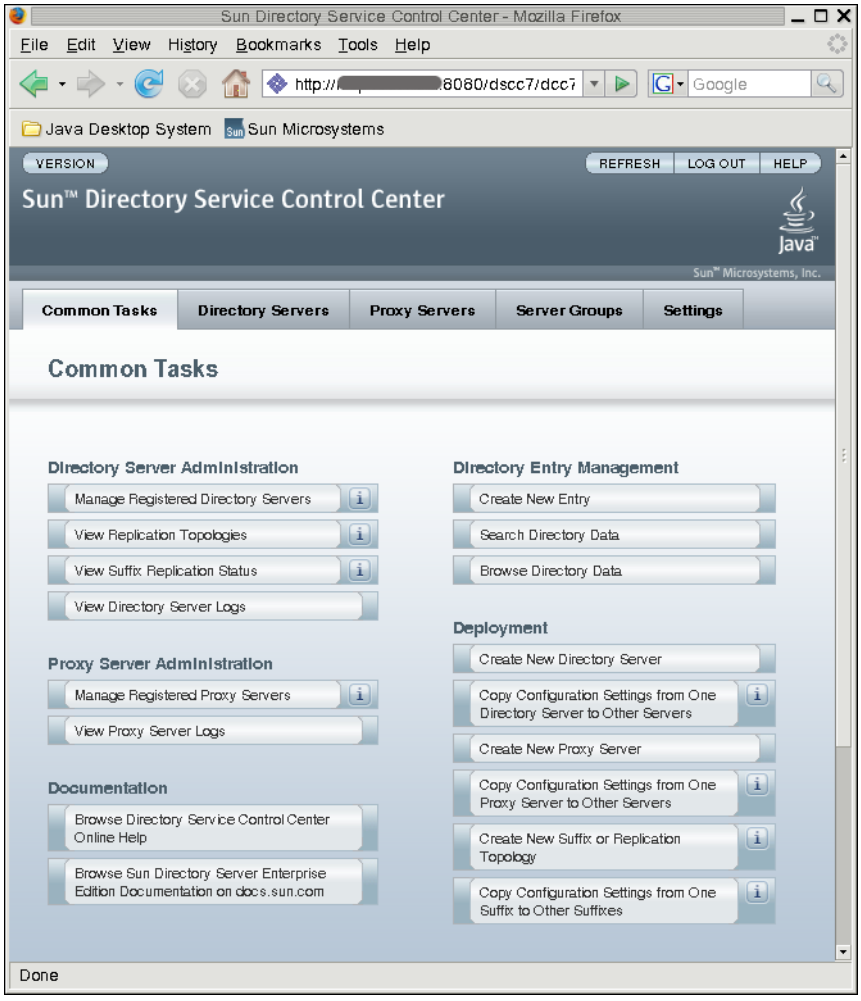


FIGURE 29–1 DSCC Common Tasks Tab

4 Navigate by using the tabs.

- The Common Tasks tab contains shortcuts to commonly used windows and wizards.
- The Directory Servers tab displays all Directory Servers managed by DSCC. To see more options for managing and configuring a particular server, click the server name.
- The Proxy Servers tab displays all Directory Proxy Servers managed by DSCC. To see more options for managing and configuring a particular server, click the server name.

Note – For instructions on how to perform tasks using DSCC, see the DSCConline help.

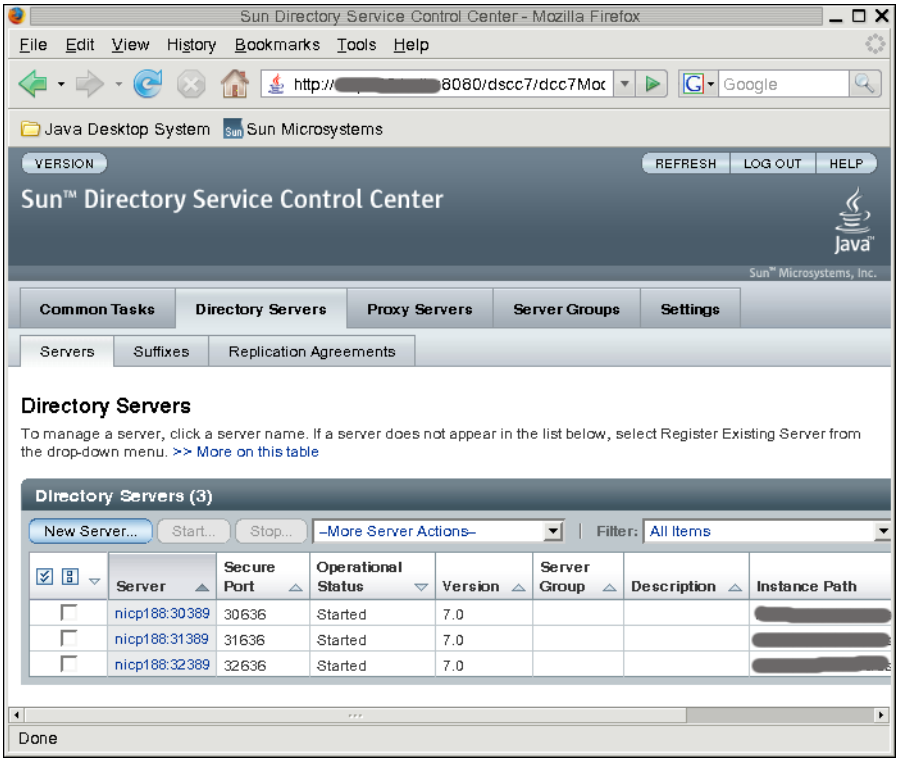


FIGURE 29–2 List of Directory Servers On the Servers Sub Tab

DSCC Command Line Interface

Following are the commands that help you work with DSCC.

dsccsetup command

The `dsccsetup` command helps in setting up DSCC. When used with appropriate subcommands, the `dsccsetup` command performs the operations such as creating the DSCC registry, initializing DSCC after installation, and registering local agents of the administration framework.

For more information, see `dsccsetup(1M)`

dsccreg command

The `dsccreg` command registers Directory Server and Directory Proxy Server instances with DSCC.

For more information, see `dsccreg(1M)`

DSCC Tabs Description

Use the tabs in DSCC to navigate the interface.

Common Tasks Tab

The Common Tasks tab (see Figure 29–1) is the first interface that you see when opening DSCC. It contains links to commonly used administrative tasks, such as searching directory data, checking logs, and managing servers.

Directory Servers Tab

The Directory Servers tab (see Figure 29–2) lists all directory servers registered in DSCC. For each server, you can see the server status and instance path, which shows where the instance is located.

When you click a server name, you see another window with a different set of tabs that relate only to that server.

Proxy Servers Tab

The Proxy Servers tab lists all the directory proxy servers that are registered in DSCC. For each server, you can see the server status and the server instance path, which shows where the instance resides.

When you click a server name, you see another window with a different set of tabs that relate only to that server.

Server Groups Tab

The Server Groups tab enables you to assign servers to groups, to make server management easier. If you have numerous servers, you can use filters to display only the servers in a certain group. You can also copy the server configuration (for example index or cache settings) from one server to all other servers in a group.

Settings Tab

This tab displays DSCC port numbers and allows you to create and delete Directory Service Managers.

DSCC Online Help

The online help provides the following:

- Context-sensitive help for the page you are currently using.
- General help for performing administration and configuration procedures using DSCC.

You can access help from most pages by clicking the Help button on the top right corner of the screen. From within a wizard, you can access help by clicking the Help tab. You can also access the online help from the Common Tasks tab.

Configuring DSCC

This section provides the following information about configuring DSCC:

▼ To Change the Common Agent Container Port Number

The default common agent container port number is 11162. The common agent container defines the DSCC agent port as `jmxmp-connector-port`. If for administrative reasons you need to use a different port number for the DSCC agent and common agent container, use the following procedure.

You cannot use DSCC to perform this task. Use the command line, as described in this procedure.

1 **For native packages installation, verify the existing port number for `jmxmp-connector-port`. You must be root to perform this operation.**

```
$ su
Password:
# cacaoadm list-params
...
jmxmp-connector-port=11162
...
```

2 **Change the DSCC agent port number.**

The common agent container must be stopped when changing the DSCC agent port number.

```
# cacaoadm stop
# cacaoadm set-param jmxmp-connector-port=new-port
# cacaoadm start
```

For the location of this command, see "Command Locations" on page 26.

3 **In DSCC, unregister your servers, and then reregister them using the new DSCC agent port number.**

In addition, when you create a new server, you must specify the non-default DSCC agent port number.

▼ To Reset the Directory Service Manager Password

To reset the Directory Service Manager password, use DSCC, as described in this procedure.

1 **Access DSCC as described in "To Access DSCC" on page 538.**

2 **Click the Settings tab, then choose Directory Service Managers.**

3 **Click the name of the Directory Service Manager for which you want to change the password.**

4 **In the properties screen, enter the new password.**

Confirm the new password by typing it again in the Confirm Password field. Click OK to save your changes.

Troubleshooting DSCC

For information about troubleshooting DSCC, see Chapter 8, "Troubleshooting DSCC Problems," in *Sun Directory Server Enterprise Edition 7.0 Troubleshooting Guide*.

Index

D

E

F

G

H

I

J

K

R

S

T

U

V