Installation Guide and Tools Reference

iPlanet™ Directory Server Resource Kit

Version 5.0

August 2001

Contents

About This Guide

This guide covers the installation of the iPlanet™ Directory Server Resource Kit (iPlanet DSRK) and contains the command-line reference for all of its tools. The LDAP APIs are part of the installation but are documented separately in the *iPlanet LDAP SDK for C Programming Guide* and the *iPlanet LDAP SDK for Java™ Programming Guide*. These programming guides are available from the same source as this document (see "Where to Find Additional Information," on page 15).

This chapter contains the following sections:

- What You Are Expected to Know
- Organization of This Guide
- Documentation Conventions
- Where to Find Additional Information

What You Are Expected to Know

Before using the tools of the iPlanet DSRK, you should be familiar with the concepts of LDAP (the Lightweight Directory Access Protocol) and common directory architecture. The books and guides described in "Where to Find Additional Information," on page 15 provide a good introduction to this background information.

In addition, you should be familiar with the command-line syntax for your shell and platform and with the deployment details of the specific LDAP directories you wish to access.

Shell Syntax

The tools of the iPlanet DSRK are command-line utilities whose functionality is available through options and parameters. You should be familiar with the syntax of commands for your particular shell and platform.

For example, you may need to specify values that contain characters with special meaning to the command-line interpreter, such as space " ", asterisk "*", backslash "\", and so forth. Most shells require you to use quotation marks around values that contain special characters. One example is the space character in DNs (distinguished names):

```
"cn=Barbara Jensen,ou=Product Development,dc=siroe,dc=com"
```

Depending on your command-line interpreter, use either single or double quote marks for this purpose. Refer to your operating system documentation for more information.

The command examples given in this book are valid for UNIX® shell environments. However, they rely on basic shell syntax and are usually applicable to all platforms with little modification.

Directory Deployment

This reference guide documents only the tools of the iPlanet DSRK. It is not intended to be a deployment, testing or maintenance guide for your directory installation. You should be aware of the following characteristics of your directory when using these tools:

- Size and number of entries.
- Directory structure and access permissions.
- Virtual attributes, class of service, and indexing.
- Usage, types of access, and access patterns.

Please refer to the *iPlanet Directory Server Deployment Guide* for information on how to determine the influence of these factors and take them into consideration.

Many other issues are further beyond the scope of the product documentation. For example, when designing performance tests, you must be able to isolate your directory from external influences, yet simulate real usage conditions. iPlanet Professional Services can help you with these and other directory deployment issues.

Organization of This Guide

This guide explains how to use each of the iPlanet DSRK tools by giving its command-line syntax, listing all options and their functionality, listing all return values when applicable, and providing examples of usage. Each tool is covered in a separate chapter and chapters for like tools are grouped within a part of the book:

Table 1 Book Parts and Chapter Contents

Title	Chapter Contents
Part 1, "Getting Started"	
Chapter 1, "Introduction to the iPlanet Directory Server Resource Kit"	Product overview.
Chapter 2, "Installation Guide"	Instructions for installing the software on all platforms.
Part 2, "Directory Access Commands"	
Chapter 3, "ldapsearch"	Command reference.
Chapter 4, "ldapmodify"	Command reference.
Chapter 5, "ldapdelete"	Command reference.
Chapter 6, "ldapcmp"	Command reference.
Part 3, "Performance Evaluation Tools"	
Chapter 7, "idsktune"	Command reference.
Chapter 8, "rsearch"	Command reference.
Chapter 9, "searchrate"	Command reference.
Chapter 10, "modrate"	Command reference.
Chapter 11, "authrate"	Command reference.
Chapter 12, "infadd"	Command reference.
Part 4, "LDIF Deployment Tools"	
Chapter 13, "dbgen.pl"	Perl script reference.
Chapter 14, "ldifgen"	Command reference.
Chapter 15, "ldifxform"	Command reference.
Chapter 16, "mmldif"	Command reference.
Chapter 17, "ldiffer.pl"	Perl script reference.

Table 1 Book Parts and Chapter Contents

Title	Chapter Contents
Part 5, "Maintenance and Debugging Tools"	
Chapter 18, "logconv.pl"	Perl script reference.
Chapter 19, "migrateSchemaTo5.pl"	Perl script reference.
Chapter 20, "searchplay"	Command reference.
Chapter 21, "viewcore"	Command reference.
Chapter 22, "dbscan"	Command reference.
Chapter 23, "Security Tools"	Links to further documentation.
Chapter 24, "Unsupported Utilities"	Brief command descriptions.
Part 6, "iPlanet LDAP Administrative Shell" (`ilash`)	
Chapter 25, "ilash Overview"	Introduces the concepts of the `ilash` shell.
Chapter 26, "ilash Command Reference"	Documents all commands within the `ilash` shell.

Chapter 26 may be printed separately to provide a complete reference to the commands within the `ilash` shell.

Documentation Conventions

This book uses the following font conventions:

- `Monospace type` is used for all command examples and code listings.
- `Monospace type` is also used within paragraph text to represent command and option names, filenames, and literal text.
- **`Monospace bold`** type shows text that a user would type at a command prompt.
- *Italic type* is used within filenames, options, and parameters to indicate a text placeholders. For example, when typing the following command, substitute your file's actual name for the *filename* placeholder:

  ```
  $ gunzip filename.tar.gz
  ```

- *Italic type* is also used within paragraph text for book titles, emphasis, and placeholder names.

Where to Find Additional Information

Nearly all tools in the iPlanet DSRK interact with iPlanet Directory Server, and the following documents are referenced throughout this guide:

- *iPlanet Directory Server Installation Guide*
- *iPlanet Directory Server Administrator's Guide*
- *iPlanet Directory Server Command, Configuration and File Reference*
- *iPlanet Directory Server Deployment Guide*

All iPlanet Directory Server documentation and directory-related documents are available in PDF and HTML formats at the following URL. The latest API documentation for the iPlanet LDAP SDKs for C and the Java programming language is also available here:

`http://docs.iplanet.com/docs/manuals/directory.html`

If you still have questions after reading this guide, the official forum for asking questions about the iPlanet DSRK is the `iplanet.server.idsrk` public newsgroup. We encourage you to contribute your LDAP client tools and directory maintenance utilities. Information about updates and new releases of the product will also be posted to this newsgroup.

The iPlanet LDAP SDKs for C and Java are also released through the Mozilla™ open source project. Visit the following site for instructions on obtaining the source code and contributing to the development of these APIs:

`http://www.mozilla.org/directory`

The Internet standards documents for LDAP and its APIs are published by the Internet Engineering Task Force (IETF). Work on these standards is ongoing, and the latest information may be found at the URL for the LDAP Extension charter:

`http://ftp.ietf.org/html.charters/ldapext-charter.html`

Many of the tools in the iPlanet DSRK are Perl scripts that rely on the `perl` interpreter. For information about the Perl language and other Perl resources, visit the following web sites:

- Comprehensive Perl Archive Network:
  `http://www.cpan.org/`
- Perl documentation:
  `http://www.perldoc.com/`

Finally, other helpful iPlanet information can be found at the following URLs:

- All online documentation for iPlanet products:
  `http://docs.iPlanet.com/`
- iPlanet Technical Support and Customer Service:
  `http://www.iPlanet.com/support/`
- iPlanet Professional Services:
  `http://www.iPlanet.com/services/professional_services_3_3.html`
- iPlanet developer information and product downloads:
  `http://developer.iplanet.com/downloads/index.jsp`
- iPlanet product data sheets:
  `http://www.iPlanet.com/products/`

Part 1

Getting Started

Chapter 1

Introduction to the iPlanet Directory Server Resource Kit

The iPlanet™ Directory Server Resource Kit (iPlanet DSRK) provides tools and APIs for deploying, accessing, tuning, and maintaining your iPlanet Directory Server. These utilities will help you implement and maintain more robust solutions based on LDAP, the Lightweight Directory Access Protocol.

The LDAP SDKs (Software Development Kits) for C and Java™ programming languages make it simple to write client applications for your directory. These APIs expose all of the functions for connecting to an LDAP directory and accessing or modifying its entries. Use them to design and integrate directory functionality into your applications at the programmatic level.

The command-line tools will help you test the performance of your Directory Server installation, debug logs and database files, and administer the contents of your directory. These tools are themselves based on the LDAP SDKs, and they were created to help iPlanet development teams to test and validate the iPlanet Directory Server.

This chapter contains the following sections:

- A Quick Installation
- iPlanet LDAP SDK for C
- iPlanet Java LDAP SDK
- Tools Reference

A Quick Installation

Follow the simple instructions in the “Installation Guide” chapter to install or update to the iPlanet Directory Server Resource Kit Version 5.0. Thanks to a simplified installer, the whole process should take less than five minutes.

The installation contains all the libraries for using the LDAP SDKs and all the executables for running the tools. No further configuration is necessary, and you can start using the software right away. To help you get started, the LDAP SDK for C contains example code, and the toolkit includes many sample Perl scripts that use the tools to perform specific maintenance operations.

iPlanet LDAP SDK for C

The iPlanet DSRK bundles version 5.04 of the iPlanet LDAP SDK for C. Use this library to write client applications in C or C++ that take full advantage of the performance of the iPlanet Directory Server. The API includes extensions that also give access to the latest features of iPlanet Directory Server 5.0.

Built around the core functions of the LDAP v2 and v3 standards, the API can be used to interact with any conforming LDAP server as well. This API conforms to IETF standard “LDAP Application Programmaning Interface,” defined by RFC 1823 and now revised by “`draft-ietf-ldapext-ldap-c-api-05`”.

The API is defined by the header files which declare all functions, data types and code values that are available in the binaries. The complete API is documented separately in the *iPlanet LDAP SDK for C Programming Guide*, available from the same source as this document. The SDK also includes sample code that demonstrates how to call most of the functions.

The iPlanet LDAP SDK for C is a binary release of the open source LDAP SDK for C source code available through `www.mozilla.org`. Updated releases are also available at:

```
http://www.iplanet.com/downloads/developer/
```

iPlanet Java LDAP SDK

The iPlanet DSRK bundles version 4.15 of the iPlanet LDAP SDK for the Java™ programming language. Equivalent in functionality to the SDK for C, Java client applications use this API to interact with LDAP directories. Whereas the Java Naming and Directory Interface™ (JNDI) provides a protocol-independent abstraction of directory services, this API exposes the LDAP-specific operations for direct access to an LDAP directory server.

Use the classes and methods of the API to develop LDAP-enabled applets or applications for the J2EE™ platform or any of the Java platforms.

The iPlanet LDAP SDK for Java conforms to the IETF standard "Java LDAP Application Program Interface," defined by draft-ietf-ldapext-ldap-java-api-15. The SDK consists of binary jar files containing all packages, classes and methods of the API. The source code is also available as open source through www.mozilla.org, and updated releases are available at:

```
http://www.iplanet.com/downloads/developer/
```

The Java API is documented separately in the *iPlanet LDAP SDK for Java Programming Guide*, also available from the same source as this document. However, the programming guide does not include all of the latest updates to the API. Please refer to the Javadoc™ pages for the latest reference information.

Tools Reference

The third component of the iPlanet Directory Server Resource Kit is the set of tools and scripts that make a directory accessible through a command-line shell. The wide range of tools can be used for simple directory access, for performance testing, and for the maintenance of directory servers. Finally, the commands that run these tools can be used to write scripts to automate all of these tasks.

This guide explains how to use each of the tools by giving their command-line reference information, usage scenarios and examples. Each tool is covered in a separate chapter and chapters for like tools are grouped within a part of the book.

Directory Access Commands

The directory access commands provide the fundamental tools for accessing a directory. Use these commands to retrieve entries, view their attributes, and make modifications. These tools are directly based upon the iPlanet LDAP SDK for C and make the functionality of this API available through their various options.

Table 1-1 Directory Access Commands

Command	Purpose
`ldapsearch`	Perform simple and complex searches to retrieve data.
`ldapmodify`	Modify the attribute values of one or more entries, or add new entries.
`ldapdelete`	Delete one or more entries given by their DN (distinguished name).
`ldapcmp`	Compare attribute values of entries in a directory.

Performance Evaluation Tools

The performance evaluation tools help you run tests to measure your server's average response time to client requests. These tools perform repeated LDAP authentication, search, add, and delete operations to simulate actual usage. Use these tools before and after reconfiguring your directory to optimize performance, and run them regularly to monitor server response as your directory size and usage evolve.

Table 1-2 Performance Evaluation Tools

Command	Purpose
`idsktune`	Optimize the OS and network settings on your system for iPlanet Directory Server.
`rsearch`	Measure the performance of search, compare, and delete operations.
`searchrate`	Measure search performance under high server loads.
`modrate`	Measure the performance of modification operations.
`authrate`	Measure the performance of establishing connections and performing authentication.
`infadd`	Measure the performance of add operations for creating new entries.

LDIF Deployment Tools

LDIF (LDAP Data Interchange Format) is the standard format for importing and exporting directory contents. The LDIF deployment tools process large LDIF files, either generating, modifying, or comparing the LDAP entries and attribute values they contain. Use these tools to generate large databases for testing, for making global directory updates offline, and for synchronizing multiple unconnected databases.

Table 1-3 LDIF Deployment Tools

Command	Purpose
`dbgen.pl`	Generate random data for tests with the performance evaluation tools.
`ldifgen`	Generate random data for tests with legacy tools.
`ldifxform`	Edits an LDIF file for global updates and extracts data for reports.
`mmldif`	Simulates a multi-master merge using LDIF files.
`ldiffer.pl`	Synchronizes differences between two directories.

Maintenance and Debugging Tools

The maintenance and debugging tools help directory administrators to interpret the log files and other trouble-shooting files. Use these tools to determine the causes of errors when they occur, as well as to perform preventive maintenance by monitoring directory usage and server statistics.

Table 1-4 Maintenance and Debugging Tools

Command	Purpose
`logconv.pl`	Helps interpret access logs and compiles usages statistics.
`migrateSchemaTo5.pl`	Helps automate the process of updating your schema for iPlanet Directory Server 5.0.
`searchplay`	Replays search operations in the directory access log.
`viewcore`	Debugs a core file.
`dbscan`	Creates text output of iPlanet Directory Server database files.
Security Tools	Manage security databases.
Unsupported Utilities	This chapter gives a brief description for each of the Perl scripts provided in the *installDir*`/unsupported/perl` directory.

iPlanet Administrative Shell

The iPlanet Administrative Shell (`ilash`) provides a complete shell environment for performing LDAP operations and managing the directory server. The `ilash` tool is based on the Tcl language and is an interpreter of Tcl scripts. Its built-in commands navigate through a directory as in a file system. Use the `ilash` tool to simplify directory access and to create powerful scripts for directory maintenance.

Chapter 2

Installation Guide

The iPlanet Directory Server Resource Kit is available on CD-ROM or from the following iPlanet download site:

```
http://www.iplanet.com/downloads/developer/
```

This chapter describes the procedure for installing the iPlanet DSRK software on the following platforms:

- Solaris Platforms
- Windows NT
- Other Platforms: AIX, HP-UX, Linux, and OSF/1

You should also read the following sections concerning all platforms:

- Files and Directories
- Release Notes URL

Solaris Platforms

The iPlanet DSRK is supported on the Solaris 2.6 and Solaris 8 platforms. There is a separate installation package for each Solaris platform, each containing native binaries for the designated version. However, the package structure and the installation procedure is identical for both.

The product requires 45 MB of disk space on the either Solaris platform and an extra 17 MB of temporary space to install from the CD-ROM. You will need twice as much temporary space if you download the zip file from the iPlanet website.

You should remove any previous versions of the Directory Server Resource Kit and the iPlanet LDAP SDKs from your machine before installing or upgrading to the iPlanet DSRK.

Installation Procedure

1. On the CD-ROM, locate the zip file for your version of the Solaris platform:

   ```
   SunOS5.8/idsrk50-SunOS5.8.zip
   SunOS5.6/idsrk50-SunOS5.6.zip
   ```

   OR download the zip file with the corresponding name from the iPlanet website and store it in an empty directory.

2. Extract the contents of the zip file with the following command:

   ```
   $ unzip idsrk50-SunOS5.x.zip -d tempDir
   ```

   Where *tempDir* is an empty directory needed only during the installation. You may omit the `-d` option when you have already downloaded the zip file to an empty directory.

3. The extracted files will contain the `setup` application, the setup configuration files, and the software in compressed format. As root, run the `setup` application:

   ```
   # tempDir/setup
   ```

4. The `setup` application will lead you through the installation of the software. When asked for input, you may press Return to accept the default, or type a new value. You may also type Control-B to go back to the previous screen or Control-C to quit without installing.

   When prompted to do so, you must type `Yes` or `yes` to accept the software license. The default is to not accept the license and exit the setup application.

   You will also be asked to choose a location for the software. The default directory is `/opt/iPlanet`. Some functionality of the iPlanet DSRK tools relies on this default directory. You may have to specify additional tool options or set your `LD_LIBRARY_PATH` if you install the software in another location.

   The software is bundled as a single component to be installed all at once. Press Return to install the single component when asked to do so.

5. After the installation of the software is complete, you no longer need the contents of the temporary directory. Use the following command to remove it and all files it contains:

   ```
   $ rm -Rf tempDir/*
   ```

Once installed, the APIs and tools of the iPlanet Directory Server Resource Kit are ready to be used. See “Files and Directories,” on page 32 for a description of the product contents.

Removing the Software

The product includes an `uninstall` application to remove the iPlanet DSRK software from your machine. Use this application before installing an upgraded version of the product.

1. If you customized any files in the software bundle and wish to keep them, you must first rename them or copy them to another location outside of the installation directory. The `uninstall` application will remove all product files by name, but it will not remove other files found under the installation directory. Any personal configuration files may thus be kept and used with an upgraded version of the software when it is installed in the same location.

2. To remove the software, run the following command as root:

   ```
   # installDir/uninstall
   ```

3. The `uninstall` application will ask you to specify the components and subcomponents to remove. The software is a single component and will be removed all at once. Press Return twice to select the single component by default and remove all iPlanet DSRK software.

   If there are no custom files to keep, the `uninstall` application will also remove the installation directory.

The `uninstall` application relies on the files in the *installDir*`/setup` directory that was created at installation. If these file are deleted or modified, the application will be unable to remove the software. In that case you may delete the software manually, after moving any customized files you wish to keep to a different location. As root, type the following command:

```
# rm -Rf installDir/*
```

The iPlanet DSRK is not bundled as a Solaris package, meaning that it is not registered with the system-wide list of packages. Therefore, removing all files will remove the iPlanet DSRK from your system.

Windows NT

The iPlanet DSRK is supported on Windows NT 4.0 Workstation or Server for the Intel platform. Service pack 5 or later is recommended. The software should also work on the Windows 2000 platform.

The product requires 21 MB of disk space and an extra 10 MB of temporary space to install from the CD-ROM. You will need twice as much temporary space if you download the zip file from the iPlanet website.

Installation Procedure

1. Log in to Windows NT as a user with `Administrator` privileges, if you have not done so already.

2. On the CD-ROM, locate the zip file for the Windows NT platform:

   ```
   WINNT4.0\idsrk50-WINNT4.0.zip
   ```

   OR download the zip file with the same name from the iPlanet website and store it in an empty folder.

3. Extract the contents of the zip file with any zip extraction utility. Extract all contents to an empty folder needed only during the installation.

   If you downloaded the zip file into an empty folder, you may extract its contents in the same folder.

4. The extracted files will contain the `setup.exe` application, the setup configuration files, and the software in compressed format. Double-click the `setup.exe` icon to begin the installation.

5. The setup application will lead you through the installation of the software. You may use the "Back" and "Next" buttons to modify any choices or the "Cancel" button to quit without installing.

   When prompted to do so, you must click "Yes" to accept the software license and continue the installation.

   You will also be asked to choose a location for the software. The default folder is `C:\opt\iPlanet`. Some functionality of the iPlanet DSRK tools relies on this default folder. You may have to specify additional tool options if you install the software in another location.

   The software is bundled as a single package to be installed all at once. This package is selected by default for installation. After you have reviewed all of your choices, click the "Install" button to copy the software files to your disk.

6. After the setup application has finished, you no longer need the contents of the temporary folder where the zip file was extracted, and you may delete it.

7. The setup program does not modify the Windows Registry. In order to do so, you should double-click on the `iDSRK.reg` icon in the new installation folder. However, you must first edit the contents of this file if you did not install the software in the default location. You must change the following line in the file to contain the location where you installed the software:

   ```
   "basedir"="c:\\opt\\iplanet"
   ```

8. The tools of iPlanet DSRK are designed to be launched from the command interpreter not from the Windows desktop. You should add the following library locations to the PATH environment variable:

   `PATH=%PATH%;`*installDir*`\lib;`*installDir*`\lib\nss\lib`

9. The `ilash` tool relies on the Tcl interpreter, version 8.2 or later, and several other tools require the Perl interpreter, version 5.005_03 or later. If you wish to use these tool you must download and install these applications. Please refer to “Tcl Resources,” on page 211 and “Where to Find Additional Information,” on page 15 for links to these applications.

When your installation is complete, see “Files and Directories,” on page 32 for a description of the product contents.

Removing the Software

The product includes an `uninst.exe` application to remove the iPlanet DSRK software from your machine. Use this application before installing an upgraded version of the product. You must be a user with `Administrator` privileges to remove the software.

1. If you customized any files in the software bundle and wish to keep them, you must first rename them or copy them to another location outside of the installation folder. The `uninst.exe` application will remove all product files by name, but it will not remove other files found under the installation folder. Any personal configuration files may thus be kept and used with an upgraded version of the software when it is installed in the same location.

2. To remove the software, double-click the `uninst.exe` icon in the installation folder.

3. The `uninst.exe` application will ask you to specify the components and subcomponents to remove. The software is a single component that is already selected by default. Click “Uninstall” to remove all iPlanet DSRK software.

   If there are no custom files to keep, the `uninst.exe` application will also remove the installation folder.

The `uninst.exe` application relies on the files in the *installDir*`/setup` folder that was created at installation. If these file are deleted or modified, the application will be unable to remove the software. In that case you may delete the installation folder manually, after moving any customized files you wish to keep to a different location.

Other Platforms

The iPlanet Directory Server Resource Kit is supported on the following platforms with the following disk space requirements:

- AIX 4.33 or later. The product requires 46 MB of disk space and an extra 20 MB of temporary space during the installation procedure.
- HP-UX B 11.0.The product requires 35 MB of disk space and an extra 15 MB of temporary space during the installation procedure.
- OSF/1 v4.0D. The product requires 51 MB of disk space and an extra 21 MB of temporary space during the installation procedure.
- Red Hat Linux 6.0 on the Intel architecture. The product requires 36 MB of disk space and an extra 17 MB of temporary space during the installation procedure.

The given temporary space requirements are for an installation from the CD-ROM. You will need twice as much temporary space if you download the zip file from the iPlanet website.

The following software installation and removal procedures may need to be tailored to the command environment available on your platform.

Installation Procedure

1. On the CD-ROM, locate the zip file for your platform:

   ```
   AIX4.3/idsrk50-AIX4.3.zip
   HP-UXB.11.00/idsrk50-HP-UXB.11.00.zip
   OSF1V4.0D/idsrk50-OSF1V4.0D.zip
   Linux2.2_x86/idsrk50-Linux2.2_x86.zip
   ```

   OR download the zip file with the corresponding name from the iPlanet website and store it in an empty directory.

2. Extract the contents of the zip file with the following command:

```
$ unzip idsrk50-SunOS5.x.zip -d tempDir
```

Where *tempDir* is an empty directory needed only during the installation. You may omit the `-d` option when you have already downloaded the zip file to an empty directory.

3. The extracted files will contain the `setup` application, the setup configuration files, and the software in compressed format. Run the `setup` application with super-user privileges:

   ```
   # tempDir/setup
   ```

4. The `setup` application will lead you through the installation of the software. When asked for input, you may press Return to accept the default, or type a new value. You may also type Control-B to go back to the previous screen or Control-C to quit without installing.

   When prompted to do so, you must type `Yes` or `yes` to accept the software license. The default is to not accept the license and exit the setup application.

   You will also be asked to choose a location for the software. The default directory is `/opt/iPlanet`. Some functionality of the iPlanet DSRK tools relies on this default directory. You may have to specify additional tool options or set your `LD_LIBRARY_PATH` if you install the software in another location.

   The software is bundled as a single component to be installed all at once. Press Return to install the single component when asked to do so.

5. After the installation of the software is complete, you no longer need the contents of the temporary directory. Use the following command to remove it and all files it contains:

   ```
   $ rm -Rf tempDir/*
   ```

Once installed, the APIs and tools of the iPlanet Directory Server Resource Kit are ready to be used. See “Files and Directories,” on page 32 for a description of the product contents.

Removing the Software

The product includes an `uninstall` application to remove the iPlanet DSRK software from your machine. Use this application before installing an upgraded version of the product.

1. If you customized any files in the software bundle and wish to keep them, you must first rename them or copy them to another location outside of the installation directory. The `uninstall` application will remove all product files by name, but it will not remove other files found under the installation directory. Any personal configuration files may thus be kept and used with an upgraded version of the software when it is installed in the same location.

2. To remove the software, run the following command with super-user privileges:

   ```
   # installDir/uninstall
   ```

3. The `uninstall` application will ask you to specify the components and subcomponents to remove. The software is a single component and will be removed all at once. Press Return twice to select the single component by default and remove all iPlanet DSRK software.

   If there are no custom files to keep, the `uninstall` application will also remove the installation directory.

The `uninstall` application relies on the files in the *installDir*`/setup` directory that was created at installation. If these file are deleted or modified, the application will be unable to remove the software. In that case you may delete the software manually, after moving any customized files you wish to keep to a different location. Run the following command with super-user privileges:

```
# rm -Rf installDir/*
```

The iPlanet DSRK is installed only by copying files, meaning that it is not registered with any system-wide list of packages or installed software. Therefore, removing all files will remove the iPlanet DSRK from your system.

Files and Directories

This guide uses the term *installDir* to refer to the installation directory, either the default or custom location where you chose to install the software. This directory contains the following files and directories containing the product components.

Table 2-1 Files and Directories of the iPlanet DSRK Installation

Directory	Contents
`bin`	Contains executable binary files for most tools in the toolkit.
`perl`	Contains Perl scripts for the remaining, Perl-based tools: `dbgen.pl`, `ldiffer.pl`, `logconv.pl`, and `migrateSchemaTo5.pl`.

Table 2-1 Files and Directories of the iPlanet DSRK Installation

Directory	Contents
`lib`	Contains binary libraries needed for running the tools. These include the same binaries as contained in the iPlanet LDAP SDK for C.
`lib/lash` `lib/tcl8.2`	Contains Tcl scripts used by the `ilash` tool.
`lib/ldapcsdk`	The base directory for the iPlanet LDAP SDK for C. Its subdirectories contain the include files and the binary libraries needed to develop and run applications using this C API.
`lib/ldapjdk`	The base directory for the iPlanet LDAP SDK for Java. It contains the jar files needed to develop and run applications using this Java API.
`lib/nss`	The base directory for the Netscape Security Services, containing subdirectories for the run-time libraries and tools binaries.
`etc`	Location of the system-wide configuration file for the `ilash` tool. The installation includes the file `system.lashconfig-sample`.
`data`	Location of the data files for the `dbgen.pl` tool.
`unsupported`	Contains sample Perl scripts and the PerLDAP tool, all provided “as-is” with no support and no endorsement.
`setup`	Contains log files from the setup process. These files are used by the `uninstall` program and should not be modified or deleted.

Release Notes URL

This guide documents the tools of the iPlanet Directory Server Resource Kit, Version 5.0, released in August 2001. Release notes are published and kept up to date at the following URL:

```
http://docs.iplanet.com/docs/manuals/dirsdk/50/relnotes.htm
```

Part 2

Directory Access Commands

Chapter 3

ldapsearch

The `ldapsearch` tool issues search requests to a directory and displays the result as LDIF text (see Appendix A, "LDAP Data Interchange Format," in the *iPlanet Directory Server Administrator's Guide*). Its many options allow you to perform different types of search operations, from simple entry retrieval to advanced searches that involve security and that manage LDAP referrals.

The `ldapsearch` tool is also provided with iPlanet Directory Server in the `/usr/iplanet/servers/shared/bin` directory. However, iPlanet DSRK and its updates include the latest version of the tool. If you use the Solaris operating environment, you may have an older version of `ldapsearch` in `/usr/bin`. Be sure your path is set to use the latest version in `/opt/iPlanet/bin/idsrk50`.

This chapter contains the following sections:

- Command Usage
- Return Values
- Command-Line Examples

Command Usage

A search involves binding and possibly authenticating to the directory server and initiating a search operation with a certain scope from a given base DN. The request includes a filter of the attribute values that must match in the entries returned. The command-line options allow you to sort the results, limit how much information is returned, control how referrals are followed, enable a secure connection, and set a time limit for the operation.

Results are displayed as LDIF text to the standard output. By default, the results contain the DN and all attributes for each entry found by the search. The results may also be reformatted through command-line options.

Syntax

The syntax of the `ldapsearch` command line has two forms:

```
ldapsearch -b "baseDN" [ options ]  "filter"  [ attributeName ... ]
ldapsearch -b "baseDN" [ options ]  -f filterFile  [ attributeName ...]
```

Where:

- *baseDN* is the base of the search, usually enclosed in double quotes (`""`) for the shell. The `-b` *baseDN* parameter may be omitted if the `LDAP_BASEDN` environment variable is set.
- *options* are the command-line options and their parameters described in the next section.
- *filter* is an RFC 2254-compliant LDAP search filter, usually in double quotes (`""`) for the shell (see “LDAP Search Filters” in Appendix B of the *iPlanet Directory Server Administrator’s Guide*).
- The *filterFile* contains one LDAP search filter per line, each one being used for a separate search. You cannot specify a filter on the command line when using the `-f` option.
- One or more *attributeNames* specifies the list of attributes and their values to be returned for each entry matching the filter. When the list of attributes is omitted, `ldapsearch` will return all attributes permitted by the access rights of the bind DN, with the exception of operational attributes.

  If you want to retrieve operational attributes, you must explicitly specify their *attributeName*. To retrieve all regular attributes in addition to operational attributes, append an asterisk (`*`) to the attribute list.

The `ldapsearch -H` command will display a usage text that briefly describes the command-line options.

Options

The `ldapsearch` command has four types of options:

- Common options.
- Input and output options.
- Options for LDAP controls.
- SSL (Secure Socket Layer) options.

The common options listed in the following table control the binding and general behavior of the `ldapsearch` command.

Table 3-1 Common Options of the `ldapsearch` Command

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. When this option is omitted, the default is `localhost`.
`-p`	*port*	Specify the port number for accessing the directory server host. The default is `389` normally and `636` when the SSL options are used.
`-D`	*bindDN*	Specify a bind DN for accessing your directory, usually in double quotes (`""`) for the shell. If the bind DN and its password are omitted, the tool will use anonymous binding. The bind DN determines what entries and attributes will appear in the search results, according to the DN's access permissions.
`-w`	*password*	Specify the password for the bind DN.
`-b`	*baseDN*	Specify the base DN for the search operation, usually in double quotes (`""`) for the shell. You may omit this option if you specify the base DN in the `LDAP_BASEDN` environment variable.
`-s`	*scope*	Specify the scope of a search. The *scope* parameter may have one of the following values: `base` - For searching only the base entry. `one` - For searching only the children of the base entry. `sub` - For searching the base entry and all its descendants. This is the default if the `-s` option is omitted.
`-f`	*filterFile*	Specify the name of a file containing filter strings. A filter file contains one or more filters, each on a separate line: `ldapsearch` will perform a separate search with each filter, in the order found in the file.
`-l`	*seconds*	Specify the maximum number of seconds to wait for a search request to complete. Regardless of the value specified here, `ldapsearch` will never wait longer than is allowed by the server's `nsslapd-timelimit` attribute, whose default is 3,600 seconds. For more information, see "nsslapd-timelimit (Time Limit)" in the *iPlanet Directory Server Command, Configuration and File Reference*.
`-V`	*version*	Specify the LDAP protocol version number to be used for the search operation, either `2` or `3`. LDAP v3 is the default; only specify LDAP v2 when connecting to servers that do not support v3.

Table 3-1 Common Options of the `ldapsearch` Command *(Continued)*

Option	Parameter	Purpose
`-Y`	*proxyDN*	Specify the proxy DN to use for the search operation, usually in double quotes (`""`) for the shell. For more information about proxy authorization, see Chapter 6, "Managing Access Control," in the *iPlanet Directory Server Administrator's Guide*.
`-a`	*aliasMode*	Specify how aliases are dereferenced when encountered in a search. The *aliasMode* parameter may have one of the following values: `never` - Aliases are never dereferenced; this is the default. `find` - Aliases are dereferenced only while finding the base DN. `search` - Aliases are dereferenced when searching entries below the base DN (but not when finding the base DN). `always` - Aliases are dereferenced both when finding the base DN and searching beneath it.
`-M`		Manage smart referrals: when referrals are part of the search results, return the actual entry containing the referral instead of the entry obtained by following the referral. For more information, see "Smart Referrals" in Chapter 5 of the *iPlanet Directory Server Deployment Guide*.
`-O`	*hopLimit*	(Capital letter O) Specify the maximum number of referral hops to follow while searching.
`-R`		Specify that referrals should *not* be followed. By default, referrals are followed automatically.
`-v`		Verbose output mode: the tool will display additional information about the search, such as the filter string and the number of results for each search.
`-n`		No-op mode: use with the `-v` option to show what the tool would do with the given input but do not actually perform the search.
`-0`	(zero)	Allow runtime library version mismatches. When this option is omitted, the default behavior is to assert that the revision number of the LDAP API is greater than or equal to that used to compile the tool. Also, if the API library and the tool have the same vendor name, the tool will also assert that the vendor version number of the API is greater than or equal to that used to compile the tool. This information is based on the contents of the `LDAPAPIInfo` structure (see the *iPlanet LDAP SDK for C Programming Guide*).
`-H`		Display the usage help text that briefly describes all options.

The input and output options given in the following table control how the `ldapsearch` results are sorted and presented.

Table 3-2 Input and Output Options of the `ldapsearch` Command

Option	Parameter	Purpose
`-1`		Omit the leading `"version: 1"` line in the LDIF output.
`-z`	*max*	Specify the maximum number of entries to return in response to a search request. Regardless of the value specified here, `ldapsearch` will never return more entries than the number allowed by the directory server's `nsslapd-sizelimit` attribute whose default is 2,000 entries. See "nsslapd-sizelimit (Size Limit)" in the *iPlanet Directory Server Command, Configuration and File Reference*. This limitation does not apply if you bind as the root DN (with `-D "cn=directory manager"`), in which case this option defaults to `0` (zero) and the size limit attribute is overridden.
`-S`	*attribute* *-attribute*	Specify an attribute for sorting the entries returned by the search. The sort criteria is alphabetical on the attribute's value or reverse alphabetical with the form *-attribute*. You may give multiple `-S` options to refine the sorting, for example: `-S sn -S givenname`. By default, the entries are not sorted.
`-i`	*locale*	Specify the character set to use for command-line input. The default is the character set specified in the `LANG` environment variable. You might want to use this option to perform the conversion from the specified character set to UTF8, thus overriding the `LANG` setting. Using this argument, you can input the bind DN, base DN, and the search filter pattern in the specified character set. The `ldapsearch` tool converts the input from these arguments before it processes the search request. For example, `-i no` indicates that the bind DN, base DN, and search filter are provided in Norwegian. This argument only affects the command-line input. If you specify a file containing a search filter (with the `-f` option), `ldapsearch` will not convert the data in the file.
`-k`	*path*	Specify the path to a directory containing conversion routines. Use these routines if you wish to specify a sorting language that is not supported by default by your directory server. For more information, see "Searching an Internationalized Directory" in Appendix B of the *iPlanet Directory Server Administrator's Guide*.
`-A`		Specify that the search retrieve only attribute names, not the attribute values. This option is useful if you just want to determine if an attribute is present for an entry.

Table 3-2 Input and Output Options of the `ldapsearch` Command *(Continued)*

Option	Parameter	Purpose
`-u`		User-friendly DNs: specify that this alternate form of the DN also be included in the output of a search result entry, in addition to the complete form that is always displayed.
`-t`		Temporary file output: each attribute of each entry in the search results will be written to a separate file in the system's temporary directory (usually `/tmp`). The standard output of the tool will include the name of the file instead of the attribute's value. When using this option, no base-64 encoding is performed on the values, regardless of the content.
`-U`		URL format (valid only with the `-t` option): when using temporary file output, the standard output of the tool will include the URL of the file instead of the attribute's value, for example: `jpegPhoto:< file:/tmp/ldapsearch-jpegPhoto-YzaOMh`
`-T`		Format the output of search results so that no line breaks are used within individual attribute values.
`-o`		Format the output of search results so that no line breaks are used within individual attribute values and that equal signs (=) are used to separate attribute names from values. This argument produces output whose format is not compliant with LDIF.
`-F`	*separator*	Format the output of search results so that the given *separator* is used between attribute names and their values. This option may be used only in conjunction with the `-o` option.
`-B`		Format the output of search results to print binary values as they are stored in the directory. When used in conjunction with `-o`, the binary data in the output will not use base-64 encoding.

The options in the following table provide advanced search controls for server-side sorting, virtual lists, and persistent searches. This functionality is available only if the server supports the corresponding LDAP controls. These options will also display any additional information that the server sends in response to the control.

Table 3-3 Options for LDAP Controls Using the `ldapsearch` Command

Option	Parameter	Purpose
`-x`		Use with the `-S` option to specify that search results be sorted on the server rather than by the `ldapsearch` command running on the client. This is useful if you want to sort according to a matching rule, as with an international search. It is usually faster to sort on the server, if that is supported, rather than on the client.
`-G`	*pattern*	Virtual list view: retrieve only a portion of all results, as determined by the index or value of the search target and the number of entries to be returned before and after the target. This option always requires the `-S` and `-x` options to specify the sorting order on the server. The pattern has two possible formats: • *entriesBefore*`:`*entriesAfter*`:`*value* - Specify the search target as the first entry in the sorted results for which the sort attribute is "greater than" or equal to the given *value*. For example, `-S sn -x -G 5:10:johnson` will return 16 entries in alphabetical order of the surname attribute: 5 less than `johnson`, the entry equal to or following `johnson`, and the 10 subsequent entries. • *entriesBefore*`:`*entriesAfter*`:`*index*`:`*count* - Specify the search target as the *index* position relative to the estimated *count*. If the *count* is `0` (zero), the index is taken as the absolute index of the target entry within the actual number of entries found. An *index* of `1` will always select the first entry in the sorted list of results. Otherwise, the target index is the first entry in slice of the list represented by the fraction *index* / *count*. For example, `-G 5:10:2:4` specifies the index closest to the beginning of the second quarter of the entire list. If the search yielded 100 entries, the target index would be 26, and this pattern would return entries 21 through 36. Give an *index* greater than the *count* to specify the last search result in the list. The number of *entriesBefore* and *entriesAfter* that are displayed may be limited by the beginning and end of the virtual list. After the results, `ldapsearch` will display the control response that gives the total count of entries in the virtual list and the actual index of the target entry. Use these values to refine the search with more accurate *index* and *count* parameters.

Table 3-3 Options for LDAP Controls Using the `ldapsearch` Command *(Continued)*

Option	Parameter	Purpose
`-C`	*pattern*	Persistent search: perform a search that keeps the connection open and displays results whenever entries matching the scope and filter of the search are added, modified, or removed. With this option, the `ldapsearch` tool will run indefinitely, and you must type Control-C to stop it. The *pattern* has the following format: `ps:`*changeType*`[:`*changesOnly*`[:`*entryChangeControls*`]]` The *changeType* determines which modifications to an entry are detected and displayed in the output. Its possible values are `add`, `delete`, `modify`, `moddn`, or `any`. The *changesOnly* parameter is an optional boolean value: specify `0`, `f`, or `false` to display the results of the search before waiting for changes. The default is 1: only changes will be displayed when they occur. By default, the tool will instruct the server to return entry change controls with the persistent search results. These controls indicate the type of operation that caused the entry to be detected by the search. The *entryChangeControls* parameter is also an optional boolean value: specify `0`, `f`, or `false` if you do not want the server to return entry change controls. In this case, you must also specify a value for the *changesOnly* parameter.

The SSL (Secure Socket Layer) options allow you to use LDAPS (LDAP over SSL) to establish a secure connection for the search. These options are valid only when LDAPS has been turned on and configured in your SSL-enabled directory server. For information on certificate-based authentication and creating a certificate database for use with LDAP clients, see Chapter 11, “Managing SSL,” in the *iPlanet Directory Server Administrator’s Guide*.

The SSL options are listed in the following table. See “Using Authentication,” on page 50 for examples using the SSL options.

Table 3-4 SSL Options of the `ldapsearch` Command

Option	Parameter	Purpose
`-Z`		Specify that SSL be used to provide a secure search operation. This option is redundant with `-P` and should no longer be used. It is kept for backwards compatibility.

Table 3-4 SSL Options of the `ldapsearch` Command *(Continued)*

Option	Parameter	Purpose
`-P`	*path*	Specify the path and filename of the client's certificate database. This file may be the same as the certificate database for an SSL-enabled version of Netscape™ Communicator, if available; for example: `-P /home/`*uid*`/.netscape/cert7.db`. When using the command on the same host as the directory server, you may use the server's own certificate database, for example: `-P /usr/iplanet/servers/slapd-`*serverID*`/alias/cert7.db`.
`-N`	*certificate*	Specify the certificate name to use for certificate-based client authentication, for example: `-N "Directory-Cert"`.
`-m`	*path*	Specify the path to the security module database. For example, `/usr/iplanet/servers/slapd-`*serverID*`/secmodule.db`. You need to specify this option only if the security module database is in a different directory from the certificate database itself.
`-K`	*keyFile*	Specify the file and path name of the client's private key database. This option may be omitted if the key database is in the location already given by the `-P` option.
`-W`	*password*	Specify the password for the client's key database given in the `-K` or `-P` options. This option is required for certificate-based client authentication.

Return Values

The `ldapsearch` tool is based on the iPlanet LDAP SDK for C, and its return values are those of the functions it uses, such as `ldap_simple_bind_s()`, `ldap_search_ext()`, and `ldap_result()`. These functions return both client-side and server-side errors and codes.

The following table shows the possible return values when the directory is hosted on iPlanet Directory Server. Other LDAP servers may send these values under different circumstances or may send different values. The directory server responding to the `ldapsearch` tool may also send other result codes in addition to those described here, for example, custom result codes from a custom plug-in.

For further information about result codes, see the *iPlanet LDAP SDK for C Programming Guide*.

Table 3-5 Return Values of the `ldapsearch` Command

Return Value	Result Code and Explanation
0 (0x00)	`LDAP_SUCCESS`: the operation was successful.
1 (0x01)	`LDAP_OPERATIONS_ERROR`: sent by the Directory Server for general errors encountered by the server when processing the request.
2 (0x02)	`LDAP_PROTOCOL_ERROR`: the search request did not comply with the LDAP protocol. The Directory Server may send this error code if it could not sort the search results or could not send sorted results.
3 (0x03)	`LDAP_TIMELIMIT_EXCEEDED`: sent by the Directory Server if the search exceeded the maximum time specified by the `-l` option.
4 (0x04)	`LDAP_SIZELIMIT_EXCEEDED`: sent by the Directory Server if the search found more results than the maximum number of results specified by the `-z` option.
10 (0x0a)	`LDAP_REFERRAL`: sent by the Directory Server if the given base DN is an entry not handled by the current server and if the referral URL identifies a different server to handle the entry.
11 (0x0b)	`LDAP_ADMINLIMIT_EXCEEDED`: sent by the Directory Server if the search found more results than the limit specified by the `lookthroughlimit` directive in the `slapd.conf` configuration file. If not specified in the configuration file, the default limit is 5000.
21 (0x15)	`LDAP_INVALID_SYNTAX`: sent by the Directory Server if your substring filter contains no value for comparison.
32 (0x20)	`LDAP_NO_SUCH_OBJECT`: sent by the Directory Server if the given base DN does not exist and if no referral URLs are available.
50 (0x32)	`LDAP_INSUFFICIENT_ACCESS`: sent by the Directory Server if the DN used for authentication does not have permission to read from the directory.
53 (0x35)	`LDAP_UNWILLING_TO_PERFORM`: sent by the Directory Server if the database is read-only.
81 (0x51)	`LDAP_SERVER_DOWN`: the LDAP server did not receive the request or the connection to the server was lost.
82 (0x52)	`LDAP_LOCAL_ERROR`: an error occurred when receiving the results from the server.
83 (0x53)	`LDAP_ENCODING_ERROR`: the request could not be BER-encoded.
84 (0x54)	`LDAP_DECODING_ERROR`: an error occurred when decoding the BER-encoded results from the server.
85 (0x55)	`LDAP_TIMEOUT`: the search exceeded the time specified by the `-l` option.

Table 3-5 Return Values of the `ldapsearch` Command *(Continued)*

Return Value	Result Code and Explanation
87 (0x57)	`LDAP_FILTER_ERROR`: an error occurred when parsing and BER-encoding a search filter specified on the command line or in a filter file.
89 (0x59)	`LDAP_PARAM_ERROR`: one of the options or parameters is invalid.
90 (0x5a)	`LDAP_NO_MEMORY`: memory cannot be allocated as needed.
91 (0x5b)	`LDAP_CONNECT_ERROR`: the specified hostname or port is invalid.
92 (0x5c)	`LDAP_NOT_SUPPORTED`: the `-V 2` option is needed to access a server that only supports LDAP v2.

Command-Line Examples

The examples in this section demonstrate common uses of the `ldapsearch` tool to access a directory. All examples assume the following context:

- You want to perform a search of all entries in the directory.
- All entries in directory are stored under `dc=siroe,dc=com`.
- The directory server has been configured to support anonymous access for search and read. Therefore, you do not have to specify any bind information in order to perform the search.
- The server is located on a machine with the given *hostname*.
- The server uses port number `389`. Because this is the default port, you do not have to specify the port number on the search request.
- SSL is enabled for the server on port `636` (the default SSL port number).

Returning All Entries

Given the context, the following command will return all entries in the directory:

```
$ ldapsearch -h hostname -b "dc=siroe,dc=com" -s sub \
             "objectclass=*"
```

The `"objectclass=*"` parameter is a search filter that matches any entry in the directory. The scope is set to the full subtree of the base DN (`-s sub`), and no attribute list is given, so all attributes for all entries will be returned.

Narrowing a Search

To narrow a search, specify a search filter enclosed in quotation marks directly on the command line. Then, you can ask to receive only those attributes that you need. For example:

```
$ ldapsearch -h hostname -b "dc=siroe,dc=com" \
             "cn=babs jensen" mail telephonenumber
```

In this example, the search will return only the `mail` and `telephonenumber` attributes of all entries with a common name (`cn`) that matches `"babs jensen"` exactly.

Searching the Root DSE Entry

The root DSE is a special entry that contains a list of all the suffixes supported by the local directory server. You can view this entry by performing a search with an empty search base (`-b ""`). You must also specify a search scope of `base` and a filter of `"objectclass=*"`, as follows:

```
$ ldapsearch -h hostname -b "" -s base "objectclass=*"
```

Searching the Schema Entry

iPlanet Directory Server stores all directory server schema in the special entry with the DN `"cn=schema"`. This entry contains information on every object class and attribute defined for your directory server.

You can examine the contents of this entry as follows:

```
$ ldapsearch -h hostname -b "cn=schema" -s base "objectclass=*"
```

Using LDAP_BASEDN

To make searching easier, you can set your search base using the `LDAP_BASEDN` environment variable. Doing this allows you to avoid specifying the search base with the `-b` option every time you use the `ldapsearch` tool. For information on how to set environment variables, see the documentation for your operating environment.

Typically, you set LDAP_BASEDN to your directory's root suffix value. Because the root suffix is the topmost entry in your directory, all searches will be able to scan the entire directory tree. For example, suppose you have set LDAP_BASEDN to dc=siroe,dc=com. Then, to search for cn=babs jensen in your directory, use the following command line:

```
$ ldapsearch -h hostname "cn=babs jensen"
```

In this example, the default scope is sub because the -s option was not used to specify the scope explicitly.

Using a Filter File

You can store search filters in a file instead of entering them on the command line. When creating a filter file, specify each search filter on a separate line. The ldapsearch command runs a separate search with each filter in the order in which the filters appear in the file. This example uses a file named myFilters that contains the following lines:

```
sn=Francis
givenname=Richard
```

Suppose search base is defined by the LDAP_BASEDN environment variable, then the following command returns all the entries that match either search filter:

```
$ ldapsearch -h hostname -f myFilters
```

In the output, ldapsearch first displays all the entries with the surname Francis, and then all the entries with the given name Richard. The two searches are independent, so an entry that matches both search criteria will be returned twice.

You can limit the set of attributes returned by specifying the attribute names that you want at the end of the search line. For example, the following ldapsearch command performs both searches, but returns only the surname and the given name attributes of each entry:

```
$ ldapsearch -h hostname -f myFilters sn givenname
```

Specifying Commas in Filters

When a DN within a search filter contains a comma as part of its value, you must escape the comma with a backslash (\). For example, to find everyone in the "Siroe Bolivia, S.A." subtree of the directory, use the following command:

```
$ ldapsearch -h hostname \
            -b "o=Siroe Bolivia\, S.A.,dc=siroe,dc=com" \
            "objectclass=*"
```

Using Authentication

There are two levels of authentication that the directory server may enforce with clients such as the `ldapsearch` tool: server authentication and client authentication. In server authentication, the server accepts connections only from clients that have a trusted certificate. In the stronger client authentication the client must sign the certificate with a password-protected private key.

To perform a search with server authentication, give only the `-P` SSL option on the command line, in addition to other common options. For example:

```
$ ldapsearch -h hostname -p 636 -b "dc=siroe,dc=com" \
            -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword \
            -P /home/bjensen/certs/cert.db \
            "givenname=Richard"
```

To perform a search with client authentication, you must give all SSL options on the command line, in addition to other common options. However, do not use the `-D` and `-w` options with client authentication, otherwise the bind operation will use the authentication credentials specified with `-D` and `-w` instead of the certificate credentials. For example:

```
$ ldapsearch -h hostname -p 636 -b "dc=siroe,dc=com" \
            -P /home/bjensen/security/cert.db -N "bjscert" \
            -K /home/bjensen/security/key.db -W KeyPassword \
            "givenname=Richard"
```

In either case, use the `-p` option to specify your directory server's SSL port. All non-SSL options retain their original meaning and may be used as necessary.

Chapter 4

ldapmodify

The ldapmodify tool edits the contents of an LDAP directory, either by adding new entries or by modifying existing ones. The tool takes update statements in LDIF as input and issues the corresponding LDAP request to the designated directory server. By placing all update statements in a file, ldapmodify can be used to process large numbers of modifications and transfer entries between directories.

The ldapmodify tool is also provided with iPlanet Directory Server in the /usr/iplanet/servers/shared/bin directory. However, iPlanet DSRK and its updates include the lastest version of the tool. If you use the Solaris operating environment, you may have an older version of ldapmodify in /usr/bin. Be sure your path is set to use the latest version in /opt/iPlanet/bin/idsrk50.

This chapter contains the following sections:

- Command Usage
- Return Values
- Command-Line Examples

Command Usage

The ldapmodify tool processes update statements, also known as change records, defined by the LDIF standard. An update statement contains the DN of the target entry for the update, the operation to perform, and any data for the entry's attributes. The operation to perform is given by the changetype keyword, and the ldapmodify tool supports the following operations:

```
add        delete        modify        modrdn        moddn        rename
```

The syntax for each of these update statements is described in “LDIF Update Statements,” in Chapter 2 of the *iPlanet Directory Server Administrator’s Guide*. Some sample update statements are given in “Command-Line Examples,” on page 59.

The `ldapmodify` tool reads any number of update statements from the standard input or from a file, and it modifies the corresponding entries according to the LDIF instructions. For each DN in the LDIF file, the tool will perform the requested LDAP operation on the designated entry.

The `ldapmodify` tool also has a special option (`-a`) for adding entries in bulk. In this case, the input should not contain the `changetype` keyword, and each of the given DNs will be added as a new entry. Using this feature, the output of the `ldapsearch` tool may be used as input to the `ldapmodify` command.

Syntax

The syntax of the `ldapmodify` command line has three forms:

```
ldapmodify [ options ]
ldapmodify [ options ] < LDIFfile
ldapmodify [ options ] -f LDIFfile
```

Where:

- *options* are the command-line options and their parameters described in “Options,” on page 53.
- *LDIFfile* is an RFC 2849-compliant LDIF text file containing update statements (see “LDIF Update Statements,” in Chapter 2 of the *iPlanet Directory Server Administrator’s Guide*) or plain LDIF entries when using the `-a` option (see Appendix A, “LDAP Data Interchange Format,” in the *iPlanet Directory Server Administrator’s Guide*).

In the first form without any *LDIFfile* input, the tool will expect you to type one or more LDIF update statements to the standard input. Once you enter all update statements and the EOF (end-of-file) marker, `ldapmodify` will process your input and perform all operations. The EOF marker is platform dependent:

- Type Control-D (**`^D`**) on most UNIX systems.
- Type Control-Z (**`^Z`**) and then press Enter on Windows NT.

The `ldapmodify -H` command will display a usage help text that briefly describes all options.

Modification Prerequisites

When modifying the contents of a directory, you must satisfy several prerequisite conditions. First, the bind DN and password used for authentication must have the appropriate permissions for the operations you perform. Many high level directory operations, such as creating a database suffix, may only be performed by the Directory Manager with a bind DN of `"cn=directory manager"`.

Then, if schema checking is active in your directory, the server will check the contents of new and modified entries against the definition of their object class in the schema. All attributes of an entry, even those not being modified, are checked against their definition and must meet the following conditions:

- The value and value type of all attributes being added or modified must conform to their definition in the entry’s object class. When this is not the case, the modification of this entry will fail.
- Attributes and values not being modified must also conform to the schema. The modification of the entry will fail even if the offending attribute is not being modified. This situation can occur if you run the directory server with schema checking turned off, remove a required attribute or set an illegal value, and then turn schema checking on. For more information, see Chapter 9, “Extending the Directory Schema,” in the *iPlanet Directory Server Administrator’s Guide*.

When a modification fails, only the operation on the faulty entry is affected, but `ldapmodify` will stop processing further input. All entries that were processed before the error was encountered will be successfully added or modified. Use the `-c` option to specify that the tool should continue processing input after any failed modification.

Finally, you must ensure the coherence of the entries in the LDIF input. Updates are performed on entries in the order they are given in the input, allowing you to manage dependencies between operations. For example, if you want to add entries to a subtree that doesn’t exist yet, your LDIF input must first give the update statement for adding the subtree entry, before the update statements for adding entries under the subtree.

Options

The `ldapmodify` command has three types of options:

- Common options.
- Input and output options.

- SSL (Secure Socket Layer) options.

The common options listed in the following table control the binding and general behavior of the `ldapmodify` command.

Table 4-1 Common Options of the `ldapmodify` Command

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. When this option is omitted, the default is `localhost`.
`-p`	*port*	Specify the port number for accessing the directory server host. The default is `389` normally and `636` when the SSL options are used.
`-D`	*bindDN*	Specify a bind DN for accessing your directory, usually in double quotes (`""`) for the shell. If the bind DN and its password are omitted, the tool will use anonymous binding. The bind DN determines what entries and attributes may be modified, according to the DN’s access permissions.
`-w`	*password*	Specify the password for the bind DN.
`-f`	*LDIFfile*	Give the name of a file containing LDIF update statements (see “LDIF Update Statements,” in Chapter 2 of the *iPlanet Directory Server Administrator’s Guide*). The tool will perform each of the update operations (add, modify, or delete) in the order given in the file. When this option is omitted, `ldapmodify` will read LDIF update statements from the standard input.
`-B`	*baseDN*	Specify the base DN when performing additions, usually in double quotes (`""`) for the shell. All entries will be placed under this suffix, thus providing bulk import functionality.
`-V`	*version*	Specify the LDAP protocol version number to be used for the modify operation, either `2` or `3`. LDAP v3 is the default; only specify LDAP v2 when connecting to servers that do not support v3.
`-Y`	*proxyDN*	Specify the proxy DN to use for the modify operation, usually in double quotes (`""`) for the shell. For more information about proxy authorization, see Chapter 6, “Managing Access Control,” in the *iPlanet Directory Server Administrator’s Guide*.
`-M`		Manage smart referrals: when they are the target of the update, modify the actual entry containing the referral instead of the entry obtained by following the referral. For more information, see “Smart Referrals” in Chapter 5 of the *iPlanet Directory Server Deployment Guide*.
`-O`	*hopLimit*	(Capital letter O) Specify the maximum number of referral hops to follow while finding an entry to modify. By default, there is no limit.

Table 4-1 Common Options of the `ldapmodify` Command *(Continued)*

Option	Parameter	Purpose
`-R`		Specify that referrals should *not* be followed. By default, referrals are followed automatically.
`-q`		Quiet output mode: the tool will not display any output about the operations it performs.
`-v`		Verbose output mode: the tool will display additional information about the operations it performs.
`-n`		No-op mode: use with the `-v` option to show what the tool would do with the given input but do not perform any operations.
`-H`		Display the usage help text that briefly describes all options.

The input and output options given in the following table control how `ldapmodify` processes input files and handles errors.

Table 4-2 Input and Output Options of the `ldapmodify` Command

Option	Parameter	Purpose
`-a`		Add entry mode: all input entries that do not contain an LDIF `changetype` statement are processed as add operations. Entries given with a `changetype` statement are processed accordingly. The add entry mode provides an easy way to add entries in LDIF. In particular, it allows you to directly add entries from the output files of the `ldapsearch` tool.
`-F`		Force the application of all updates, regardless of the replica status.
`-i`	*locale*	Specify the character set to use for the `-f` *LDIFfile* or standard input. The default is the character set specified in the `LANG` environment variable. You might want to use this option to perform the conversion from the specified character set to UTF8, thus overriding the `LANG` setting.
`-k`	*path*	Specify the path to a directory containing conversion routines. These routines are used if you wish to specify a locale that is not supported by default by your directory server. For more information, see “Searching an Internationalized Directory” in Appendix B of the *iPlanet Directory Server Administrator’s Guide*.

Table 4-2 Input and Output Options of the `ldapmodify` Command *(Continued)*

Option	Parameter	Purpose
`-b`		Handle binary files: the `ldapmodify` tool will scan every attribute value in the input to determine whether it is a valid file reference, and if so, it will use the contents of the file as the attribute's value. This option is used to input binary data for an attribute, such as a JPEG image. For example, the corresponding LDIF input would be: • `jpegPhoto: /tmp/photo.jpg` (on a UNIX platform) • `jpegPhoto: c:\tmp\photo.jpg` (on Windows NT) This option allows you to directly process entries from the output files of the `ldapsearch` tool when it uses the `-t` option. The `ldapmodify` tool also supports the LDIF `:<` *URL* notation for directly including file contents. For example: • `jpegPhoto:< file:///tmp/photo.jpg` (on all platforms) If all of your input entries use this notation, you do not need to specify the `-b` option.
`-A`		Non-ASCII mode: display non-ASCII values, in conjunction with the `-v` option.
`-c`		Continuous mode: errors are reported but the `ldapmodify` tool will continue processing input and performing operations. When this option is omitted, the default is to quit after reporting an error.
`-e`	*errorFile*	Invalid update statements in the input will be copied to the *errorFile* for debugging. Use with the `-c` option to correct errors when processing large LDIF input.

The SSL (Secure Socket Layer) options listed in the following table allow you to use LDAPS (LDAP over SSL) to establish a secure connection for the update operation. These options are valid only when LDAPS has been turned on and configured in your SSL-enabled directory server. For information on certificate-based authentication and creating a certificate database for use with LDAP clients, see Chapter 11, "Managing SSL" in the *iPlanet Directory Server Administrator's Guide*.

Only the `-P` option is required for server authentication. For the more secure client authentication, the `-P`, `-N`, `-K` and `-W` options are required. See "Using Authentication," on page 61 for examples using the SSL options.

Table 4-3 SSL Options of the `ldapmodify` Command

Option	Parameter	Purpose
`-Z`		Specify that SSL be used to provide a secure modify operation. This option is redundant with `-P` and should no longer be used. It is kept for backwards compatibility.
`-P`	*path*	Specify the path and filename of the client's certificate database. This file may be the same as the certificate database for an SSL-enabled version of Netscape Communicator, if available; for example: `-P /home/`*uid*`/.netscape/cert7.db`. When using the command on the same host as the directory server, you may use the server's own certificate database, for example: `-P /usr/iplanet/servers/slapd-`*serverID*`/alias/cert7.db`.
`-N`	*certificate*	Specify the certificate name to use for certificate-based client authentication, for example: `-N "Directory-Cert"`.
`-m`	*path*	Specify the path to the security module database. For example, `/usr/iplanet/servers/slapd-`*serverID*`/secmodule.db`. You need to specify this option only if the security module database is in a different directory from the certificate database itself.
`-K`	*keyFile*	Specify the file and path name of the client's private key database. This option may be omitted if the key database is in the location already given by the `-P` option.
`-W`	*password*	Specify the password for the client's key database given in the `-K` or `-P` options. This option is required for certificate-based client authentication.

Return Values

The `ldapmodify` tool is based on the iPlanet LDAP SDK for C and its return values are those of the functions it uses, such as `ldap_simple_bind_s()`, `ldap_add_ext_s()`, `ldap_modify_ext_s()`, and `ldap_delete_ext_s()`. These functions return both client-side and server-side errors and codes.

The following table shows the possible return values when the directory is hosted on an iPlanet Directory Server. Other LDAP servers may send these values under different circumstances or may send different values. The directory server responding to the `ldapmodify` tool may also send other result codes in addition to those described here, for example, custom result codes from a custom plug-in.

For further information about result codes, see the *iPlanet LDAP SDK for C Programming Guide*.

Table 4-4 Return Values of the `ldapmodify` Command

Return Value	Result Code and Explanation
0 (0x00)	`LDAP_SUCCESS`: the operation was successful.
1 (0x01)	`LDAP_OPERATIONS_ERROR`: sent by the Directory Server for general errors encountered by the server when processing the request.
2 (0x02)	`LDAP_PROTOCOL_ERROR`: the modify request did not comply with the LDAP protocol. The Directory Server may set this error code in the results for a variety of reasons, such as encountering an error when decoding the BER-encoded request.
10 (0x0a)	`LDAP_REFERRAL`: sent by the Directory Server if the specified DN is an entry not handled by the current server and if the referral URL identifies a different server to handle the entry.
16 (0x10)	`LDAP_NO_SUCH_ATTRIBUTE`: sent by the Directory Server if the attribute that you want to modify (add, replace, or delete) does not exist.
19 (0x13)	`LDAP_CONSTRAINT_VIOLATION`: sent by the Directory Server when improperly modifying the `userpassword` attribute, for example if the new value is shorter than the allowed minimum length.
20 (0x14)	`LDAP_TYPE_OR_VALUE_EXISTS`: sent by the Directory Server when attempting to add an attribute to an entry in which the attribute already exists with the given value.
21 (0x15)	`LDAP_INVALID_SYNTAX`: sent by the Directory Server if your client is modifying the schema entry and no object class or attribute type is specified.
32 (0x20)	`LDAP_NO_SUCH_OBJECT`: sent by the Directory Server if the entry that you want to modify or delete does not exist.
50 (0x32)	`LDAP_INSUFFICIENT_ACCESS`: sent by the Directory Server if the DN used for authentication does not have permission to write to the entry.
53 (0x35)	`LDAP_UNWILLING_TO_PERFORM`: sent by the Directory Server when: - The directory is read-only. - Attempting to add attributes to the special directory configuration entry. - Attempting to modify attributes in the special schema entry.
65 (0x41)	`LDAP_OBJECT_CLASS_VIOLATION`: sent by the Directory Server if the modified entry does not comply with the directory schema (for example, if one or more required attributes are not specified).
67 (0x43)	`LDAP_NOT_ALLOWED_ON_RDN`: sent by the Directory Server if the modified entry no longer contains attributes for each DN component.

Table 4-4 Return Values of the `ldapmodify` Command *(Continued)*

Return Value	Result Code and Explanation
68 (0x44)	`LDAP_ALREADY_EXISTS`: sent by the Directory Server if the DN of the entry that you want add is already present in the directory.
81 (0x51)	`LDAP_SERVER_DOWN`: the LDAP server did not receive the request or the connection to the server was lost.
82 (0x52)	`LDAP_LOCAL_ERROR`: an error occurred when receiving the results from the server.
83 (0x53)	`LDAP_ENCODING_ERROR`: BER-encoding the request is not possible.
84 (0x54)	`LDAP_DECODING_ERROR`: an error occurred when decoding the BER-encoded results from the server.
89 (0x59)	`LDAP_PARAM_ERROR`: one of the options or parameters is invalid.
90 (0x5a)	`LDAP_NO_MEMORY`: memory cannot be allocated as needed.
91 (0x5b)	`LDAP_CONNECT_ERROR`: the specified hostname or port is invalid.
92 (0x5c)	`LDAP_NOT_SUPPORTED`: the `-V 2` option is needed to access a server that only supports LDAP v2.

Command-Line Examples

The examples in this section demonstrate common uses of the `ldapmodify` tool to update the contents a directory. All examples assume the following context:

- The given bind DN has the permission to perform all operations on the selected entries.
- The directory server is located on a machine with the given *hostname*.
- The server uses port number `389`. Because this is the default port, you do not have to specify the port number on the search request.
- SSL is enabled for the server on port `636` (the default SSL port number).

Adding an Entry

This example uses the `-a` option for bulk addition, so the the `changetype: add` is not needed in the input. Instead, the input contains standard LDIF entries to be added. The input file is called `newEntry.ldif` and contains only one entry to add:

```
dn: cn=Pete Minsky,ou=People,dc=siroe,dc=com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Pete Minsky
givenName: Pete
sn: Minsky
ou: People
ou: Marketing
uid: peterm
```

To perform this addition, launch the `ldapmodify` tool with the `-a` option and specify the input file with the `-f` option:

```
$ ldapmodify -h hostname -a -f newEntry.ldif \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword
```

Modifying an Entry

The update statement for a modification includes statements for specifying the attributes to change and their new values. See “LDIF Update Statements,” in Chapter 2 of the *iPlanet Directory Server Administrator’s Guide* for a description of this syntax.

In this example, the `mofifyEntry.ldif` file includes statements for adding a new attribute and modifying an existing one. The line with a single dash (-) is a separator for multiple modifications in the same entry:

```
dn: cn=Pete Minsky,ou=People,dc=siroe,dc=com
changetype: modify
add: telephonenumber
telephonenumber: (408) 555-2468
-
replace: uid
uid: pminsky
```

To perform the operation, launch the `ldapmodify` tool and specify the filename on the command line.

```
$ ldapmodify -h hostname -f mofifyEntry.ldif \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword
```

Deleting an Entry

The update statement for a deletion requires only the DN and the changetype. This example shows how to enter this information as standard input:

```
$ ldapmodify -h hostname \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword

dn: cn=Pete Minsky,ou=People,dc=siroe,dc=com
changetype: delete
^D
```

Using Authentication

There are two levels of authentication that the directory server may enforce on clients such as the ldapmodify tool: server and client authentication. In server authentication, the server accepts only connections from clients that have a trusted certificate. In the stronger client authentication, the certificate is not assumed to be trusted, so the client must sign it with a password-protected private key.

To run the ldapmodify tool with server authentication, give only the -P option on the command line, in addition to other options. For example:

```
$ ldapmodify -h hostname -p 636 -f LDIFfile \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword \
             -P /home/bjensen/certs/cert.db
```

To perform an update with client authentication, you must give all SSL options on the command line, in addition to other options. However, do not use the -D and -w options with client authentication, otherwise the bind operation will use the authentication credentials specified with -D and -w instead of the certificate credentials. For example:

```
$ ldapmodify -h hostname -p 636 -f LDIFfile \
             -P /home/bjensen/security/cert.db -N "bjscert" \
             -K /home/bjensen/security/key.db -W KeyPassword
```

In either case, use the -p option to specify your directory server's SSL port. All other options remain the same and may be used as necessary.

Chapter 5

ldapdelete

The `ldapdelete` tool is a simple command for deleting entries in an LDAP directory. In its simplest form, it takes distinguished names (DNs) on the command line or from the standard input and deletes the corresponding entries. It can also take input from a file for bulk processing and has options for configuring advanced features such as SSL security.

The `ldapdelete` tool is also provided with iPlanet Directory Server in the `/usr/iplanet/servers/shared/bin` directory. However, iPlanet DSRK and its updates include the lastest version of the tool. If you use the Solaris operating environment, you may have an older version of `ldapdelete` in `/usr/bin`. Be sure your path is set to use the latest version in `/opt/iPlanet/bin/idsrk50`.

This chapter contains the following sections:

- Command Usage
- Return Values
- Command-Line Examples

Command Usage

The `ldapdelete` command binds to the given directory server and issues a delete command for each entry given by a DN in the input. In order to delete an entry in a directory server, the DN used for binding and authentication must have the permission to delete the given entries.

Only leaf entries, which do not have any children, may be deleted from a directory. For example, when deleting a subtree representing an organization unit, you must first delete all the entries it contains before deleting the entry representing the organizational unit.

When deleting DNs listed in a file, the tool will process each delete operation separately, in the order they are given in the file. Therefore, DNs representing leaf entries must be listed before the DNs for their parent entries.

Syntax

The syntax of the `ldapdelete` command line has four forms:

```
ldapdelete [ options ]
ldapdelete [ options ] "DN" ...
ldapdelete [ options ] < DNfile
ldapdelete [ options ] -f DNfile
```

Where:

- *options* are the command-line options and their parameters described in the next section.
- *DN* ... is a space-separated list of DNs to delete. Each DN should be enclosed in double quote marks ("") for the shell interpreter. The list of DNs is not required if you give the DNs as standard input or in a *DNfile*.
- *DNfile* is a text file containing one DN per line. Do not use quote marks in this file because each line is taken literally.

In the first form without any *DN* input, the tool will expect you to type one or more DNs to the standard input. Once you enter all DNs and the EOF (end-of-file) marker, `ldapdelete` will process your input and perform all operations. The EOF marker is platform dependent:

- Type Control-D (**`^D`**) on most UNIX systems.
- Type Control-Z (**`^Z`**) and then press Return on Windows NT.

The `ldapdelete -H` command will display a usage help text that briefly describes all options.

Options

The `ldapdelete` command has three types of options:

- Common options.
- Input and output options.

- SSL (Secure Socket Layer) options.

The common options listed in the following table control the binding and general behavior of the `ldapdelete` command.

Table 5-1 Common Options of the `ldapdelete` Command

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. When this option is omitted, the default is `localhost`.
`-p`	*port*	Specify the port number for accessing the directory server host. The default is `389` normally and `636` when the SSL options are used.
`-D`	*bindDN*	Specify a bind DN for accessing your directory, usually in double quotes (`""`) for the shell. If the bind DN and its password are omitted, the tool will use anonymous binding. The bind DN determines what entries may be deleted, according to the DN's access permissions.
`-w`	*password*	Specify the password for the bind DN.
`-f`	*DNfile*	Give the name of a file containing the DNs of entries to be deleted. The DNs should be listed one per line in this file, in the order in which they must be deleted. When this option is omitted, `ldapdelete` will read DNs directly from the standard input.
`-V`	*version*	Specify the LDAP protocol version number to be used for the delete operation, either `2` or `3`. LDAP v3 is the default; only specify LDAP v2 when connecting to servers that do not support v3.
`-Y`	*proxyDN*	Specify the proxy DN to use for the delete operation, usually in double quotes (`""`) for the shell. For more information about proxy authorization, see Chapter 6, "Managing Access Control," in the *iPlanet Directory Server Administrator's Guide*.
`-M`		Manage smart referrals: when they are the target of the operation, delete the actual entry containing the referral instead of the entry obtained by following the referral. For more information, see "Smart Referrals" in Chapter 5 of the *iPlanet Directory Server Deployment Guide*.
`-O`	*hopLimit*	(Capital letter O) Specify the maximum number of referral hops to follow while finding an entry to delete. By default, there is no limit.
`-R`		Specify that referrals should *not* be followed. By default, referrals are followed automatically.
`-v`		Verbose output mode: the tool will display additional information about the operations it performs.

Table 5-1 Common Options of the `ldapdelete` Command *(Continued)*

Option	Parameter	Purpose
`-n`		No-op mode: use with the `-v` option to show what the tool would do with the given input but do not perform the delete operations.
`-H`		Display the usage help text that briefly describes all options.

The input and output options given in the following table control how `ldapdelete` processes input files and handles errors.

Table 5-2 Input and Output Options of the `ldapdelete` Command

Option	Parameter	Purpose
`-i`	*locale*	Specify the character set to use for command-line input. The default is the character set specified in the `LANG` environment variable. You might want to use this option to perform the conversion from the specified character set to UTF8, thus overriding the `LANG` setting. Using this argument, you can input the bind DN and the target DNs in the specified character set. The `ldapdelete` tool converts the input from these arguments before it processes the search request. For example, `-i no` indicates that the bind DN and target DNs are provided in Norwegian. This option affects only the command-line input, that is, if you specify a file containing DNs (with the `-f` option), `ldapsearch` will not convert the data in the file.
`-k`	*path*	Specify the path to a directory containing conversion routines. These routines are used if you wish to specify a locale that is not supported by default by your directory server. For more information, see “Searching an Internationalized Directory” in Appendix B of the *iPlanet Directory Server Administrator’s Guide*.
`-c`		Continuous mode: errors are reported but the `ldapdelete` tool will continue processing input and performing operations. When this option is omitted, the default is to quit after reporting an error.

The SSL (Secure Socket Layer) options listed in the following table allow you to use LDAPS (LDAP over SSL) to establish a secure connection for the delete operation. These options are valid only when LDAPS has been turned on and configured in your SSL-enabled directory server. For information on certificate-based authentication and creating a certificate database for use with LDAP clients, see Chapter 11, "Managing SSL" in the *iPlanet Directory Server Administrator's Guide*.

Only the `-P` option is required for server authentication. For the more secure client authentication, the `-P`, `-N`, `-K` and `-W` options are required. See "Using Authentication," on page 71 for examples using the SSL options.

Table 5-3 SSL Options of the `ldapdelete` Command

Option	Parameter	Purpose
`-Z`		Specify that SSL be used to provide a secure delete operation. This option is redundant with `-P` and should no longer be used. It is kept for backwards compatibility.
`-P`	*path*	Specify the path and filename of the client's certificate database. This file may be the same as the certificate database for an SSL-enabled version of Netscape Communicator, if available; for example: `-P /home/`*uid*`/.netscape/cert7.db`. When using the command on the same host as the directory server, you may use the server's own certificate database, for example: `-P /usr/iplanet/servers/slapd-`*serverID*`/alias/cert7.db`.
`-N`	*certificate*	Specify the certificate name to use for certificate-based client authentication, for example: `-N "Directory-Cert"`.
`-m`	*path*	Specify the path to the security module database. For example, `/usr/iplanet/servers/slapd-`*serverID*`/secmodule.db`. You need to specify this option only if the security module database is in a different directory from the certificate database itself.
`-K`	*keyFile*	Specify the file and path name of the client's private key database. This option may be omitted if the key database is in the location already given by the `-P` option.
`-W`	*password*	Specify the password for the client's key database given in the `-K` or `-P` options. This option is required for certificate-based client authentication.

Return Values

The ldapdelete tool is based on the iPlanet LDAP SDK for C and its return values are those of the functions it uses, such as ldap_simple_bind_s() and ldap_delete_ext_s(). These functions return both client-side and server-side errors and codes.

The following table shows the possible return values when the directory is hosted on an iPlanet Directory Server. Other LDAP servers may send these values under different circumstances or may send different values. The directory server responding to the ldapdelete tool may also send other result codes in addition to those described here, for example, custom result codes from a custom plug-in.

For further information about result codes, see the *iPlanet LDAP SDK for C Programming Guide*.

Table 5-4 Return Values of the ldapdelete Command

Return Value	Result Code and Explanation
0 (0x00)	LDAP_SUCCESS: the operation was successful.
1 (0x01)	LDAP_OPERATIONS_ERROR: sent by the Directory Server for general errors encountered by the server when processing the request.
2 (0x02)	LDAP_PROTOCOL_ERROR: the delete request did not comply with the LDAP protocol. The Directory Server may send this error code in the results for a variety of reasons, such as encountering an error when decoding the BER-encoded request.
10 (0x0a)	LDAP_REFERRAL: sent by the Directory Server if the specified DN is an entry not handled by the current server and if the referral URL identifies a different server to handle the entry.
32 (0x20)	LDAP_NO_SUCH_OBJECT: sent by the Directory Server if the entry that you want deleted does not exist and if no referral URLs are available.
50 (0x32)	LDAP_INSUFFICIENT_ACCESS: sent by the Directory Server if the DN used for authentication does not have permission to write to the entry.
53 (0x35)	LDAP_UNWILLING_TO_PERFORM: sent by the Directory Server if the database is read-only.
66 (0x42)	LDAP_NOT_ALLOWED_ON_NONLEAF: sent by the Directory Server if the entry that you want deleted has entries beneath it in the directory tree (in other words, if this entry is a parent entry to other entries).
81 (0x51)	LDAP_SERVER_DOWN: the LDAP server did not receive the request or the connection to the server was lost.

Table 5-4 Return Values of the ldapdelete Command *(Continued)*

Return Value	Result Code and Explanation
82 (0x52)	LDAP_LOCAL_ERROR: an error occurred when receiving the results from the server.
83 (0x53)	LDAP_ENCODING_ERROR: BER-encoding the request is not possible.
84 (0x54)	LDAP_DECODING_ERROR: an error occurred when decoding the BER-encoded results from the server.
89 (0x59)	LDAP_PARAM_ERROR: one of the options or parameters is invalid.
90 (0x5a)	LDAP_NO_MEMORY: memory cannot be allocated as needed.
91 (0x5b)	LDAP_CONNECT_ERROR: the specified hostname or port is invalid.
92 (0x5c)	LDAP_NOT_SUPPORTED: the -V 2 option is needed to access a server that only supports LDAP v2.

Command-Line Examples

The examples in this section demonstrate common uses of the ldapdelete tool. All examples assume the following context:

- The given bind DN has the permission to perform delete operations on the selected entries.
- The directory server is located on a machine with the given *hostname*.
- The server uses port number 389. Because this is the default port, you do not have to specify the port number on the search request.
- SSL is enabled for the server on port 636 (the default SSL port number).

Specifying Commas in DNs

The simplest usage of the tool is to specify the target DNs on the command line, making sure to enclose them in double quote marks for the shell. In addition, special characters such as commas must be escaped with a backslash (\) when they appear in components of a DN on the command line.

In this example, the user bjensen wishes to delete an entry in the "Siroe Bolivia, S.A." subtree of the directory:

```
$ ldapdelete -h hostname \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword \
             "cn=Lucia Fuentes,ou=People,o=Siroe Bolivia\,S.A."
```

Using the Standard Input

Using the `-v` and `-c` options, you can enter DNs interactively through the standard input. In this example, the user `bjensen` wishes to remove an entry but enters the DN incorrectly at first:

```
$ ldapdelete -h hostname -v -c \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword

ldapdelete: started Thu Jun 14 11:34:17 2001
ldap_init( host, 389 )

cn=Pete Mimsky,ou=People,dc=siroe,dc=com

deleting entry cn=Pete Mimsky,ou=People,dc=siroe,dc=com
ldap_delete: No such object

cn=Pete Minsky,ou=People,dc=siroe,dc=com

deleting entry cn=Pete Minsky,ou=People,dc=siroe,dc=com
entry removed

^D
```

Using a DN File

For bulk operations, list all of the DNs to delete in a text file. Specify each DN on a separate line in the file. The `ldapdelete` command will perform a delete operation on each entry, in the order in which they appear in the file. For example:

```
cn=Pete Minsky,ou=People,dc=siroe,dc=com
cn=Sue Jacobs,ou=People,dc=siroe,dc=com
```

Then specify this *DNfile* on the command line with the `-f` option. Use the `-c` option so that bulk processing will continue even if errors are encountered.

```
$ ldapdelete -h hostname -c -f DNfile \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword
```

Using Authentication

There are two levels of authentication that the directory server may enforce on clients such as the `ldapdelete` tool: server and client authentication. In server authentication, the server only accepts connections from clients that have a trusted certificate. In the stronger client authentication, the certificate is not assumed to be trusted, so the client must sign it with a password-protected private key.

To run the `ldapdelete` tool with server authentication, give only the `-P` option on the command line, in addition to other options. For example:

```
$ ldapdelete -h hostname -p 636 -f DNfile \
             -D "uid=bjensen,dc=siroe,dc=com" -w bindPassword \
             -P /home/bjensen/certs/cert.db
```

To perform an update with client authentication, you must give all SSL options on the command line, in addition to other options. However, do not use the `-D` and `-w` options with client authentication, otherwise the bind operation will use the authentication credentials specified with `-D` and `-w` instead of the certificate credentials. For example:

```
$ ldapdelete -h hostname -p 636 -f DNfile \
             -P /home/bjensen/security/cert.db -N "bjscert" \
             -K /home/bjensen/security/key.db -W KeyPassword
```

In either case, use the `-p` option to specify your directory server's SSL port. All other options remain the same and may be used as necessary.

Chapter 6

ldapcmp

The `ldapcmp` tool compares the contents of a single entry or of an entire subtree that is present in two directories. It detects entries that do not appear in both directories and any difference between attributes in those that do. This tool supports the common options of the LDAP commands, such as managing referrals, handling locales, and providing SSL-based security.

This chapter contains the following sections:

- Command Usage
- Return Values
- Command-Line Examples

Command Usage

A comparison involves performing a search operation on both directories and analyzing the differences in the results. Each search returns all attributes for all entries in the given scope of the given base DN. The `ldapcmp` tool then compares these search results and reports the following differences in its output:

- Entries that only appear in the first directory:

  `1only:` *DN*

- Entries that only appear in the second directory:

  `2only:` *DN*

- Entries whose DN appears in both directories but whose attributes or attribute values are different:

```
matchingDN
different: missingAttributeName
   1 or 2: attributeValueWherePresent
different: differingAttributeName
        1: valueInDirectory1
        2: valueInDirectory2
```

Syntax

The `ldapcmp` command has the following syntax:

```
ldapcmp -h host1 -p port1 [ -h host2 -p port2 ] -b "baseDN" [ options ]
```

Where:

- *host1, port1, host2, port2* are the hostnames and port numbers for accessing the two directories you wish to compare. The first host and port correspond to directory `1` in the output, the second to directory `2`. If the second host and port are omitted, port `389` on the `localhost` will be used by default for directory `2`.
- *baseDN* is the base of the comparison, usually enclosed in double quotes (`""`) for the shell. The `-b` *baseDN* parameter may be omitted if the `LDAP_BASEDN` environment variable is set.
- *options* are the command-line options and their parameters described in the next section. Except for the host and port options, all other options must apply to both directories being compared. For example, both directory servers must accept the same certificate when using the security options.

The `ldapcmp -H` command will display a usage help text that briefly describes all options.

Options

The `ldapcmp` command has three types of options:

- Common options.
- Input and output options.
- SSL (Secure Socket Layer) options.

The common options listed in the following table control the binding and general behavior of the `ldapcmp` command.

Table 6-1 Common Options of the `ldapcmp` Command

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of a directory server. This option may be given twice to specify the two target directories for the comparison. If it is only given once, the default for the second host is `localhost`. If it is not specified at all, the default is `localhost` for both.
`-p`	*port*	Specify the port number for accessing a directory server host. This option may be given twice to specify the port for each directory server. When either occurrence is omitted, the default is `389` normally and `636` when the SSL options are used. Note that the first occurrence of this option specifies the port for the first host, even if it appears *after* the second hostname on the command line.
`-D`	*bindDN*	Specify a bind DN for accessing both directories, usually in double quotes (`""`) for the shell. If the bind DN and its password are omitted, the tool will use anonymous binding. The bind DN determines what entries and attributes will appear in the comparison results, according to the DN's search permissions.
`-w`	*password*	Specify the password for the bind DN.
`-b`	*baseDN*	Specify the base DN for the comparison, usually in double quotes (`""`) for the shell. You may omit this option if you specify the base DN in the `LDAP_BASEDN` environment variable.
`-s`	*scope*	Specify the scope of the comparison. Use this option to restrict the number of entries being compared. The *scope* parameter may have one of the following values: `base` - For comparing only the base entry. `one` - For comparing only the children of the base entry. `sub` - For comparing the base entry and all its descendants. This is the default if the `-s` option is omitted.
`-V`	*version*	Specify the LDAP protocol version number to be used for search operations, either `2` or `3`. LDAP v3 is the default; only specify LDAP v2 when connecting to servers that do not support v3.
`-Y`	*proxyDN*	Specify the proxy DN to use for search operations, usually in double quotes (`""`) for the shell. For more information about proxy authorization, see Chapter 6, "Managing Access Control," in the *iPlanet Directory Server Administrator's Guide*.
`-M`		Manage smart referrals: when they are part of the comparison searches, return the actual entry containing the referral instead of the entry obtained by following the referral. For more information, see "Smart Referrals" in Chapter 5 of the *iPlanet Directory Server Deployment Guide*.

Table 6-1 Common Options of the `ldapcmp` Command *(Continued)*

Option	Parameter	Purpose
`-O`	*hopLimit*	(Capital letter O) Specify the maximum number of referral hops to follow when performing comparison searches.
`-R`		Specify that referrals should *not* be followed. By default, referrals are followed automatically during comparison searches.
`-v`		Verbose output mode: the tool will display additional information about binding to the directory servers, searching the two directories, and comparing the search results.
`-n`		No-op mode: use with the `-v` option to show what the tool would do with the specified input but do not actually perform the searches.
`-H`		Display the usage help text that briefly describes all options.

The input and output options given in the following table control how the `ldapcmp` results are sorted and presented.

Table 6-2 Input and Output Options of the `ldapcmp` Command

Option	Parameter	Purpose
`-i`	*locale*	Specify the character set to use for command-line input. The default is the character set specified in the `LANG` environment variable. You might want to use this option to perform the conversion from the specified character set to UTF8, thus overriding the `LANG` setting. Using this argument, you can input the bind DN in the specified character set. The `ldapcmp` tool converts the input from these arguments before it processes the search request. For example, `-i no` indicates that the bind DN and attribute names are provided in Norwegian.
`-k`	*path*	Specify the path to a directory containing conversion routines. These routines are used if you wish to specify a sorting language that is not supported by default by your directory server. For more information, see “Searching an Internationalized Directory” in Appendix B of the *iPlanet Directory Server Administrator’s Guide*.

The SSL (Secure Socket Layer) options listed in the following table allow you to use LDAPS (LDAP over SSL) to establish secure connections for the comparison. These options are valid only when LDAPS has been turned on and configured in your SSL-enabled directory server. For information on certificate-based authentication and creating a certificate database for use with LDAP clients, see Chapter 11, “Managing SSL” in the *iPlanet Directory Server Administrator’s Guide*.

Table 6-3 SSL Options of the `ldapcmp` Command

Option	Parameter	Purpose
`-Z`		Specify that SSL be used to provide secure search operations. This option is redundant with `-P` and should no longer be used. It is kept for backwards compatibility.
`-P`	*path*	Specify the path and filename of the client’s certificate database. This file may be the same as the certificate database for an SSL-enabled version of Netscape Communicator, if available; for example: `-P /home/`*uid*`/.netscape/cert7.db`. When using the command on the same host as the directory server, you may use the server’s own certificate database, for example: `-P /usr/iplanet/servers/slapd-`*serverID*`/alias/cert7.db`.
`-N`	*certificate*	Specify the certificate name to use for certificate-based client authentication, for example: `-N "Directory-Cert"`. Both of the directory servers must recognize this certificate to perform the comparison.
`-m`	*path*	Specify the path to the security module database. For example, `/usr/iplanet/servers/slapd-`*serverID*`/secmodule.db`. You need to specify this option only if the security module database is in a different directory from the certificate database itself.
`-K`	*keyFile*	Specify the file and path name of the client’s private key database. This option may be omitted if the key database is in the location already given by the `-P` option.
`-W`	*password*	Specify the password for the client’s key database given in the `-K` or `-P` options. This option is required for certificate-based client authentication.

Return Values

The ldapcmp tool is based on the iPlanet LDAP SDK for C and its return values are those of the functions it uses, such as ldap_simple_bind_s(), ldap_search_ext(), and ldap_result(). These functions return both client-side and server-side errors and codes.

The following table shows the possible return values when the directories are hosted on iPlanet Directory Servers. Other LDAP servers may send these values under different circumstances or may send different values. The directory servers responding to the ldapcmp tool may also send other result codes in addition to those described here, for example, custom result codes from a custom plug-in.

For further information about result codes, see the *iPlanet LDAP SDK for C Programming Guide*.

Table 6-4 Return Values of the ldapcmp Command

Return Value	Result Code and Explanation
0 (0x00)	LDAP_SUCCESS: the operation was successful.
1 (0x01)	LDAP_OPERATIONS_ERROR: sent by the Directory Server for general errors encountered by the server when processing the request.
2 (0x02)	LDAP_PROTOCOL_ERROR: the Directory Server may send this error code in the results for a variety of reasons, such as encountering an error when decoding the BER-encoded request.
3 (0x03)	LDAP_TIMELIMIT_EXCEEDED: sent by the Directory Server if a comparison search exceeded the default time limit for an operation on the server.
4 (0x04)	LDAP_SIZELIMIT_EXCEEDED: sent by the Directory Server if a comparison search found more results than the default maximum allowed by the server.
10 (0x0a)	LDAP_REFERRAL: sent by the Directory Server if the given base DN is an entry not handled by either server and if the referral URL identifies a different server to handle the entry.
11 (0x0b)	LDAP_ADMINLIMIT_EXCEEDED: sent by the Directory Server if a comparison search found more results than the limit specified by the lookthroughlimit directive in the slapd.conf configuration file. If not specified in the configuration file, the default limit is 5000.
32 (0x20)	LDAP_NO_SUCH_OBJECT: the given base DN cannot be found in both Directory Servers and if no referral URLs are available.
50 (0x32)	LDAP_INSUFFICIENT_ACCESS: sent by the Directory Server if the DN used for authentication does not have permission to read from the directory.

Table 6-4 Return Values of the `ldapcmp` Command *(Continued)*

Return Value	Result Code and Explanation
81 (0x51)	`LDAP_SERVER_DOWN`: either of the LDAP servers did not respond to the comparison search or a connection was lost.
82 (0x52)	`LDAP_LOCAL_ERROR`: an error occurred when receiving the results from either server.
83 (0x53)	`LDAP_ENCODING_ERROR`: the request could not be BER-encoded.
84 (0x54)	`LDAP_DECODING_ERROR`: an error occurred when decoding the BER-encoded results from either server.
89 (0x59)	`LDAP_PARAM_ERROR`: one of the options or parameters is invalid.
90 (0x5a)	`LDAP_NO_MEMORY`: memory cannot be allocated as needed.
91 (0x5b)	`LDAP_CONNECT_ERROR`: a specified hostname or port is invalid.
92 (0x5c)	`LDAP_NOT_SUPPORTED`: the `-V 2` option is needed to access a server that only supports LDAP v2.

Command-Line Examples

The examples in this section demonstrate common uses of the `ldapcmp` tool to compare two directories. All examples assume the following context:

- All entries in the directories are stored under `dc=siroe,dc=com`.
- The directory server has been configured to support anonymous access for search and read. Therefore, you do not have to specify any bind information in order to perform the search.
- The servers are located on the machines named *host1* and *host2*.
- The servers both use port number `389`. Because this is the default port, you do not have to specify the port number on the search request.

Comparing Two Directories

By specifying the root DN as the base DN, `ldapcmp` will search all entries of both directories. The output of the following command will show you *all* differences between directory contents:

```
$ ldapcmp -h host1 -h host2 -b "dc=siroe,dc=com"
```

You should have some idea of the size and differences between your directories before comparing them. Comparing two directories is useful for finding small difference between directories. The output will be very large and not very helpful if all entries are completely different. In this case, narrow the comparison by specifying the base DN of a similar subtree in both directories.

Comparing Entries

The `ldapcmp` tool can also be used to compare single entries. This operation is much quicker than comparing a subtree because the searches are faster and only a single comparison needs to be performed. If you know the DN of the entry to compare, use it as the base DN on the command line, for example:

```
$ ldapcmp -h host1 -h host2 -s base -b \
          "cn=Pete Minsky,ou=People,dc=siroe,dc=com"
```

Using LDAP_BASEDN

To simplify the command line, you can set the base DN using the `LDAP_BASEDN` environment variable. Doing this allows you to avoid specifying the search base with the `-b` option every time you use the `ldapcmp` tool. For information on how to set environment variables, see the documentation for your operating system.

For example, suppose you have set `LDAP_BASEDN` to `dc=siroe,dc=com`. Then to compare your directories on two different hosts, use the following command:

```
$ ldapcmp -v -h host1 -h host2
```

Specifying the `-v` option for verbose output is helpful because the base DN being used will be displayed in the output for verification.

Comparing Directory Configurations

The Directory Server configuration information is stored as entries in the directory, and you may use the `ldapcmp` tool to compare how your Directory Servers are configured. The following command line will compare the root DSE of two Directory Servers, such as the extensions and controls they support:

```
$ ldapcmp -h host1 -h host2 -b ""
```

Because some configuration information is host- and directory-specific, the previous command will always display some differences. Another source of configuration information is the schema used by your directories. The following command will compare two directory schemas:

```
$ ldapcmp -h host1 -h host2 -b "cn=schema"
```

Schemas can be very large, and comparisons between them are useful only if they are known to have small differences. For example, you can see if a master schema has been customized in different ways for separate directories.

Part 3

Performance Evaluation Tools

Chapter 7

idsktune

The default operating system and network settings on many platforms are not suitable for high performance directory services. Tuning is the process of modifying these settings for optimal performance of both directory clients and directory servers.

The `idsktune` tool helps automate this task by checking for necessary patches and suggesting the optimal kernel and TCP/IP settings for running iPlanet Directory Server. This tool does not modify the system; it only provides suggested settings for the system administrator to make these changes manually.

You may run `idsktune` and apply its guidelines even before installing iPlanet Directory Server. In order to get accurate measurements from performance tests, you should run `idsktune` before your tests and modify your system accordingly.

NOTE The `idsktune` tool is available only on UNIX platforms. Suggested system and registry settings for the Windows NT platform are described in Chapter 2, “Computer System Requirements” of the *iPlanet Directory Server Installation Guide*.

This chapter contains the following sections:

- System Tuning
- Command Usage
- Sample Output

System Tuning

The idsktune tool gathers information about the operating system, kernel, and TCP stack in order to make tuning recommendations.

OS and Kernel Settings

The tool displays current OS version numbers and patch information, along with any further recommended patches. It also verifies disk and memory availability and warns you of any deficiencies.

Specifically, idsktune verifies and reports on the following settings:

- Up-to-date OS and Kernel versions:
    - Solaris, OSF/1, and Red Hat Linux version numbers
    - Solaris kernel build date
    - Solaris, HP-UX, and AIX patches
- Sufficient memory and disk space:
    - Physical memory size
    - Swap space size or swap partition size
    - Memory resource limits
    - File descriptor resource limits
- Scheduler settings:
    - Maximum threads per user (OSF/1)
    - Maximum threads per process (HP-UX)
    - Maximum files (HP-UX)

TCP Settings

The idsktune tool reads the current settings of your system's TCP module and makes recommendations for changes. The tool does not perform any of these modifications. Instead, it displays the command line for the ndd tool, which the system administrator should use to set the parameter. The ndd tool is available on the Solaris platform: use an equivalent tool if you are tuning a different platform.

The system administrator should consider the local network conditions and other application needs when modifying TCP settings. In general, however, the recommendations will optimize performance whether the machine is dedicated to the directory server or shared with other applications.

The idsktune tool verifies and makes recommendations on the following settings:

- Listen backlog queue size
- tcbhashsize,tcbhashnum and tcp_msl (OSF/1)
- sominconn and somaxconn
- ipport_userreserved_min
- tcp_close_wait_interval and tcp_time_wait_interval
- tcp_keepalive_interval
- tcp_max_listen
- tcp_conn_request_max
- tcp_conn_req_max_q and tcp_conn_req_max_q0
- tcp_rexmit_interval_initial
- net.inet.ip.portrange.hifirst (Linux) and tcp_smallest_anon_port
- tcp_slow_start_initial
- net.inet.tcp.delayed_ack (Linux) and tcp_deferred_ack_interval
- link_speed on /dev/hme (Solaris platform)

The idsktune tool verifies all of the settings for the Solaris 2.6 and Solaris 8 operating systems that are described in the "TCP Tuning" section in Chapter 2 of the *iPlanet Directory Server Installation Guide*.

Further Information

Basic and advanced information about tuning your system is available through the following books and websites:

- *Sun Performance and Tuning: Java and the Internet* (ISBN 0-13-095249-4)
- *Solaris Performance Administration* (ISBN 0-07-011768-3)
- "Solaris 2.x - Tuning Your TCP/IP Stack and More" (http://www.sean.de/Solaris/tune.html)

- "Solaris Tunable Parameters Reference Manual" (`http://docs.sun.com:80/ab2/coll.709.2/SOLTUNEPARAMREF/`)
- "Tuning Tru64 UNIX for Internet Servers" (`http://www.tru64unix.compaq.com/faqs/publications/internet/TITLE.HTM`)
- `sys-check` tool for Tru64 (OSF/1) UNIX (`http://www.tru64unix.compaq.com/sys_check/sys_check.html`)

Command Usage

Although this command should be run as `root` to get a full report on all settings, most settings are verified when it is run as any user. However, you must be `root` if you wish to modify any of the OS, kernel, or TCP settings based on the `idsktune` recommendations.

Syntax

The `idsktune` command has the following syntax:

```
idsktune [-v | -D | -q | -c | -\?]
```

Options

The following table describes the command-line options:

Table 7-1 Command-Line Options for the `idsktune` Tool

Option	Parameter	Purpose
`-v`		Version: gives the build date identifying the version of the tool.
`-D`		Debug mode: the output includes the commands it runs internally, preceded by the "`DEBUG`" heading.
`-q`		Quiet mode: the output includes tuning recommendations, but OS version statements are omitted.
`-c`		Client-specific tuning: the output includes tuning recommendations for running a directory client application, but server-specific recommendations are omitted.
`-\?`		Display the usage help text that briefly describes all options.

Sample Output

The following output is an example of running idsktune on an untuned Solaris 8 system. This is the default output displayed when no options are used.

Code Example 7-1 Sample Output of the idsktune Command

```
# /opt/iPlanet/bin/idsrk50/idsktune
iPlanet Directory Server system tuning analysis
version 27-MAR-2001.
Copyright 2001 Sun Microsystems, Inc.

NOTICE : System is usparc-sun-solaris5.8 (SUNW,Ultra-5_10)
(1 processor).

NOTICE : Patch 109137-01 is not installed.

NOTICE : Patch 109320-01 is not installed.

NOTICE : Patch 108974-02 is not installed.

NOTICE : Patch 108977-01 is not installed.

NOTICE : Patch 108968-02 is not installed.

NOTICE : Patch 108975-02 is not installed.

NOTICE : Patch 108528-01 is not installed.

NOTICE : Patch 108875-07 is not installed.

NOTICE : Patch 108652-13 is not installed.

NOTICE : Solaris patches can be obtained from
http://sunsolve.sun.com or your Solaris support representative.

ERROR  : Only 128MB of physical memory is available on the system.
1024MB is the recommended minimum.

ERROR  : There is 128MB of physical memory but only 114MB of swap
space.

WARNING: The tcp_close_wait_interval is set to 240000 milli-
seconds (240 seconds). This value should be reduced to allow for
more simultaneous connections to the server.  A line similar to
the following should be added to the /etc/init.d/inetinit file:
ndd -set /dev/tcp tcp_time_wait_interval 30000

NOTICE : The tcp_conn_req_max_q value is currently 128, which
will limit the value of listen backlog which can be configured.
It can be raised by adding to /etc/init.d/inetinit, after any adb
command, a line similar to:
ndd -set /dev/tcp tcp_conn_req_max_q 1024
```

Code Example 7-1 Sample Output of the `idsktune` Command *(Continued)*

```
NOTICE : The tcp_keepalive_interval is set to 7200000 milli-
seconds (120 minutes). This may cause temporary server congestion
from lost client connections.

NOTICE : The tcp_keepalive_interval can be reduced by adding the
following line to /etc/init.d/inetinit:
ndd -set /dev/tcp tcp_keepalive_interval 600000

NOTICE : The NDD tcp_rexmit_interval_initial is currently set to
3000 milliseconds (3 seconds). This may cause packet loss for
clients on Solaris 2.5.1 due to a bug in that version of Solaris.
If clients are not using Solaris 2.5.1, no problems should occur.

NOTICE : If the directory is service is intended only for LAN or
private high-speed WAN environment, this interval can be reduced
by adding to /etc/init.d/inetinit:
ndd -set /dev/tcp tcp_rexmit_interval_initial 500

NOTICE : The NDD tcp_smallest_anon_port is currently 32768.  This
allows a maximum of 32768 simultaneous connections.  More ports
can be made available by adding a line to /etc/init.d/inetinit:
ndd -set /dev/tcp tcp_smallest_anon_port 8192

WARNING: tcp_deferred_ack_interval is currently 100 milliseconds.
This will cause Solaris to insert artificial delays in the LDAP
protocol. It should be reduced during load testing.
This line can be added to the /etc/init.d/inetinit file:
ndd -set /dev/tcp tcp_deferred_ack_interval 5

WARNING: There are only 1024 file descriptors available, which
limits the number of simultaneous connections.  Additional file
descriptors, up to 65536, are available by adding to /etc/system
a line such as:
set rlim_fd_max=4096

NOTICE : / partition has less space available, 213MB, than the
largest allowable core file size of 242MB. A daemon process which
dumps core could cause the root partition to be filled.

ERROR  : The above errors MUST be corrected before proceeding.
```

Chapter 8

rsearch

The `rsearch` (repeated search) tool is a multi-threaded LDAP client program that measures the performance of LDAP search, compare, modify, delete, and authentication operations. It performs operations continuously and computes an average operation rate at regular intervals.

This tool is written in C using the iPlanet LDAP SDK for C and may be used to measure the performance of any LDAP directory. As with all measures of performance, results depend upon many factors, such as machine load, network traffic and directory configuration, and should be analyzed accordingly. In order to obtain accurate performance measurements, both client and server machines should be dedicated to the performance test and properly tuned (see Chapter 7, “idsktune”).

This chapter contains the following sections:

- Command Usage
- Sample Output
- Command-Line Examples

Command Usage

The `rsearch` command launches a number of threads that perform synchronous operations on the given directory server. The threads are simple loops that perform the same operation over and over as quickly as possible.

At regular intervals (every 10 seconds by default) the tool displays the statistics collected for operations completed during the elapsed interval. It shows the average number of operations per thread, the total number of operations, the number of operations per second, and the inverse in milliseconds per operation.

The default operation is to perform a search with the filter given on the command line. In order to simulate more realistic directory usage, you may provide a filter file with multiple filter strings that the tool will select from randomly.

The `rsearch` tool will also measure the performance of compare, modify, delete, and authentication operations. These require you to provide a file containing the DNs or UIDs for entries that will be randomly selected for performing each operation. The output for these operations contains the same information as for searches.

Syntax

The `rsearch` command has the following syntax:

```
rsearch -D "bindDN" -w password -s "suffix" -f "filter" [ options ]
```

Where:

- *bindDN* and *password* are the bind credentials with sufficient permissions to search or modify the directory.
- *suffix* is the base DN for the search or modify operations.
- *filter* is an RFC 2254-compliant LDAP search filter (see “LDAP Search Filters” in Appendix B of the *iPlanet Directory Server Administrator’s Guide*). The filter may use the `%s` syntax to include strings from a file given with the `-i` option.
- *options* are the command-line options and their parameters described in the next section.

Command-line parameters such as DNs and filters should be enclosed in double quotes ("...") if they contain special characters for the shell. When the required options (`-D`, `-w`, `-s`, and `-f`) are not needed for a given operation, they may be given as an empty string ("").

The `rsearch -\?` command will display a usage help text that briefly describes all options.

Options

The `rsearch` options and parameters are described in the following table.

Table 8-1 Command-Line Options for the `rsearch` Tool

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. The default is `localhost`.
`-p`	*port*	Specify the port number when accessing the directory server host. The default is `389`.
`-D`	*bindDN*	Specify a bind DN for all operations, usually in double quotes (`""`) for the shell. Depending on its access permissions, the given bind DN may affect authentication and search performance.
`-w`	*password*	Specify the password for the bind DN.
`-s`	*suffix*	Specify the suffix to use as the base DN for all operations, usually in double quotes (`""`) for the shell.
`-f`	*filter*	Specify a filter for search operations, usually in double quotes (`""`) for the shell. This option can be used with an input file, for example, the filter `cn=%s` will use a random string from a file specified by the `-i` option.
`-i`	*filterFile*	Give the name of the file containing filter strings. The filter will be selected randomly from this file. See “Filter File Format,” on page 95 for more information.
`-S`	*scope*	Specify the scope of a search. The *scope* parameter may have one of the following values: `0` - For searching only the base entry. `1` - For searching one level below the base entry. `2` - For searching all levels below the entry. This is the default.
`-A`	*attributes*	A comma separated list of attribute names to specify which attribute values will be returned by searches.
`-N`		No operation: using this flag, the tool will only bind to the server, instead of binding and performing a search, which is the default. The `-c`, `-d`, and `-m` options will be ignored when this flag is used.
`-b`		Perform a bind before every operation (and unbind afterwards).
`-u`		Do not unbind from the sever, just close the connection.
`-L`		Set the linger mode on the TCP socket to avoid too many leftover closed connections.
`-y`		Set the `TCP_NODELAY` mode on the TCP socket.
`-t`	*threads*	Specify the number of threads that `rsearch` will use. Use the `-v` option for verbose output including measurements from each thread. The default is a single thread.

Table 8-1 Command-Line Options for the `rsearch` Tool *(Continued)*

Option	Parameter	Purpose
`-j`	*seconds*	Specify the sampling interval, in seconds; the default is 10. `rsearch` repeats the given operation as many times as possible during the interval and prints results after each interval elapses.
`-T`	*seconds*	Specify the time limit, in seconds, after which `rsearch` will display the cumulated average and stop. When this option is not specified, `rsearch` will run until its process is terminated.
`-V`		Alternate output at every interval: display only the rate and cumulated average rate of operations per thread.
`-v`		Verbose output at every interval: gives the measurements from each thread, including the minimum and maximum operation times observed, as well as the average for all threads.
`-q`		Quiet output mode: the measurements for each interval will not be displayed. If `-T` is specified, only the final average will be shown.
`-B`	*UIDfile* or *DNfile*	Specify the file needed for input to the compare, delete, modify, and authentication operations. Must be used in conjunction with the `-c`, `-d`, `-m` or `-x` flags, respectively, to specify the operation. See "DN and UID File Formats," on page 95 for more information.
`-c`		Perform compare operations on the `uid` attribute of entries chosen randomly from a DN file specified by the `-B` option. Comparisons are set up to be randomly true or false and verified accordingly.
`-d`		Perform delete operations on entries chosen randomly from a DN file specified by the `-B` option. Note that `rsearch` does not replace these entries, meaning that delete operations will cause an error when a DN is randomly selected the second time. This will not affect `rsearch`, but the statistics will be skewed unless the entries are otherwise restored or replaced.
`-m`		Perform modify operations on the `description` attribute, assuming it is not indexed. The entries to be modified are chosen randomly from a DN file specified by the `-B` option.
`-M`		Perform modify operations on the `telephonenumber` attribute, assuming that it is indexed. The entries to be modified are chosen randomly from a DN file specified by the `-B` option.
`-x`		Perform bind operations using a DN and its UID as a password for authentication. The DN is selected randomly from a UID or DN file specified by the `-B` option.
`-\?`		Display the usage help text that briefly describes all options.

Filter File Format

The -i option is followed by the name of a text file containing filter strings. Each line of the file is taken to be a filter string and may contain spaces, as follows:

```
Filter string 1
Filter string 2
...
```

When using a filter file, you must use the -f option with a filter containing the %s placeholder. When performing each search, the placeholder will be textually replaced by a filter string randomly selected from the filter file.

The filter may contain only one placeholder, for example, -f "cn=%s". Alternatively, you may specify -f "%s" on the command line and give complete filters in the filter file. This will allow you to perform more complex searches.

When using a filter file, it should contain the appropriate strings for searching in your directory. For example, you may want to measure exact string matching searches, searches that fail, or compound search expressions. Also the number of filter strings will determine how often the same search is repeated: few strings imply that search results are more likely to be retrieved from the directory server's cache.

DN and UID File Formats

The -B option specifies a file containing either DN and UID values or UID values alone, as shown in the following table.

Table 8-2 DN or UID File Formats

DN File Format	UID File Format
DN: dn_string UID: uid_string DN: dn_string2 UID: uid_string2 ...	UID: uid_string UID: uid_string2 ...

In these formats, the DN: and UID: keywords must be followed by white space. Everything after that space is taken literally; do not use quotes unless you want them to appear in the string.

DN files are used as input to the compare, modify, and delete operations. Both DN and UID files may be used as input for binding when performing authentication and search operations (see “Measuring Bind and Authentication Operations,” on page 97, and “Random DN Bind and Search Rate,” on page 99).

When used for binding by specifying the `-x` option, the two file formats are treated differently. If you use the DN file format, authentication for the bind will use the DN as the bind DN and the UID as the password. If you have a file containing only UIDs, `rsearch` will use its `-D` *bindDN* and `-w` *password* parameters to bind and search for an entry with the given UID. It will find the DN of that entry, and then bind again with this DN, using the UID as the password.

Sample Output

```
Rate: 648.00/thr (129.60/sec = 7.7160msec/op ), total: 1296 (2 thr)
Rate: 645.00/thr (129.00/sec = 7.7519msec/op ), total: 1290 (2 thr)
Rate: 642.50/thr (128.50/sec = 7.7821msec/op ), total: 1285 (2 thr)
...
Final Average rate: 130.74/sec = 7.6488msec/op, total:  1285
```

The given `Rate` is the average number of operation per thread over the elapsed interval (10 seconds by default). Use the `-j` *seconds* option to specify a different interval. All data on an output line concerns only the elapsed interval. Use the `-V` option to display a running average of operations per thread per interval.

If you specify a time limit with the `-T` *seconds* option, the final average rate is computed and displayed; otherwise, the command runs until you terminate it. This average is totaled over all threads, giving the absolute operation rate and its inverse, the average time to complete each operation.

Command-Line Examples

The following sections give examples of `rsearch` command usage for measuring the performance of various scenarios. You will need to adapt these examples to your environment:

- The *hostname* and *port* placeholders should be replaced with the hostname and port number of your directory. When significant, the *suffix* should represent the contents of the directory tree or subtree you wish to test.
- For meaningful results, thread numbers and time limit options should be scaled according to your directory’s expected load.

- You will also need to provide DN and UID files that correspond to the contents of your directory (see "DN and UID File Formats," on page 95).

Measuring Bind and Authentication Operations

The following examples will measure bind and authentication performance in your LDAP directory. These commands perform only bind and unbind operations. They do not perform any searches, except when part of the binding procedure.

In all of the bind and authentication examples, the `-s` *suffix* and `-f` *filter* options have no effect because no search is performed. They are given as empty strings (`""`) because they are required on the command line.

Anonymous Bind Rate

```
$ rsearch -h hostname -p port -s "" -f "" \
          -D "" -w "" -N -b -L -T 100
```

This command will bind anonymously (`-D "" -w ""`), repeat binding with no other operations (`-N -b`), avoid opening too many connections (`-L`), use a single thread (no `-t` option), display statistics every 10 seconds (no `-j` option), and finish in about 100 seconds (`-T 100`).

Random DN Authentication Rate

```
$ rsearch -h hostname -p port -s "" -f "" -D "" -w "" \
          -B DNfile -x -N -b -L
```

This command will bind repeatedly, each time as a random DN found in the DN file (`-B` *DNfile* `-x`), use the DN's UID from that file as the password, repeat binding with no other operations (`-N -b -L`), and run indefinitely (no `-T` option).

The `-D ""` and `-w ""` options are required on the command line but have no effect.

Random UID Authentication Rate

```
$ rsearch -h hostname -p port -s "" -f "" \
          -D "" -w "" -B UIDfile -x -N -b -L
```

This command is a typical authentication scenario where the client must first find the DN corresponding to a UID before binding. In each bind sequence, this command will select a random UID from the given UID file (`-B` *UIDfile* `-x`), bind anonymously (`-D "" -w ""`) to find the DN of the corresponding entry, and then bind as this DN using the UID for authentication. It will repeat the binding sequence without performing searches until the process is killed (`-N -b -L`).

You may need to set the `-D` and `-w` options for valid authentication if your directory does not allow anonymous binding.

Root DN Bind Rate

```
$ rsearch -h hostname -p port -s "" -f "" \
          -D "cn=Directory Manager" -w password -N -b -L -T 100
```

This command will bind as root DN (`-D "cn=Directory Manager" -w` *password*), and repeat binding with no other operations (`-N -b -L`) for about 100 seconds (`-T 100`).

Measuring Search Operations

The following examples will measure search performance in your directory server. All of the searches are performed within a single bind, so the results are those of the search operation alone, assuming factors such as machine load and network traffic remain constant.

In these examples, you will need to provide the suffix and filter strings for valid entries in your test directory (see "Filter File Format," on page 95). Filter files may also contain different kinds of filter strings that may give different performance results:

- Strings containing wildcards will give measurements that reflect substring search rate.
- Strings without wildcards will give measurements that reflect the search rate for exact matches.
- Filter strings may also include operators other than equality, such as ranges that are greater than (>=) or less than (<=) a given value, or approximate spelling searches (~=).

Simple Search

```
$ rsearch -h hostname -p port -s "dc=Siroe,dc=com" \
          -D "bindDN" -w password -f "cn=*jones*"
```

This command will bind once as the given DN (`-D` *bindDN* `-w` *password*) and search repeatedly for entries under the `"dc=Siroe,dc=com"` suffix that contain the `"cn=*jones*"` substring. Because every search uses the same filter and returns the same entries, entry caching in the directory server will influence results. In all of the following examples, the use of a filter file helps provide more realistic measurements.

Search Rate Using Anonymous Bind

```
$ rsearch -h hostname -p port -s "dc=Siroe,dc=com" \
          -D "" -w "" -f "sn=%s" -i filterFile
```

This command will perform a single anonymous bind (`-D "" -w ""`) and search repeatedly for random surnames taken from the filter file (`-f "sn=%s" -i` *filterFile*).

Search Rate Using DN Bind

```
$ rsearch -h hostname -p port -s "dc=Siroe,dc=com" \
          -D "bindDN" -w password -f "sn=%s" -i filterFile
```

This command will bind once as the given DN (`-D` *bindDN* `-w` *password*) and search repeatedly for entries matching random surnames taken from the filter file (`-f "sn=%s" -i` *filterFile*).

Specific Attribute Search Rate

```
$ rsearch -h hostname -p port -s "dc=Siroe,dc=com" \
          -D "" -w "" -f "sn=%s" -i filterFile -A ginenName,mail
```

This command will retrieve only the `givenName` and `mail` attributes of entries matching random surnames taken from the filter file (`-f "sn=%s" -i` *filterFile*).

Measuring Bind and Search Operations

The following commands are similar to the search examples, except that `rsearch` will bind before every search operation and unbind afterward. These examples mimic the usual behavior of directory clients and thus measure the average time taken to serve a directory client. Again, if filter files contain strings with wildcards, performance will reflect substring search rates, otherwise it will reflect exact search rates.

To further resemble actual search operations, multiple threads simulate load on the server. Set the thread option to model the expected load on your directory.

Anonymous Bind and Search Rate

```
$ rsearch -h hostname -p port -s "dc=Siroe,dc=com" \
          -D "" -w "" -f "sn=%s" -i filterFile -b -L -t 10
```

This command will create 10 threads (`-t 10`), each of which will repeatedly bind, search, and unbind (`-b`), while avoiding too many open connections (`-L`). It always uses anonymous binding (`-D "" -w ""`) and searches for random surnames (`-f "sn=%s" -i` *filterFile*) in entries under the `"dc=Siroe,dc=com"` suffix.

Random DN Bind and Search Rate

```
$ rsearch -h hostname -p port -s "dc=Siroe,dc=com" -D "" -w "" \
          -B DNfile -x -f "sn=%s" -i filterFile -b -L -t 10
```

This command will create 10 threads (`-t 10`), each of which will repeatedly bind, search for entries under the `"dc=Siroe,dc=com"` suffix, and unbind (`-b`), while avoiding too many open connections (`-L`). It always binds with a DN randomly selected from the DN file (`-B` *DNfile* `-x`) and use the DN's UID from that file as the password (see "DN and UID File Formats," on page 95). The `-D ""` and `-w ""` options are required on the command line but have no effect.

Random UID Bind and Search Rate

```
$ rsearch -h hostname -p port -s "dc=Siroe,dc=com" -D "" -w "" \
          -B UIDfile -x -f "sn=%s" -i filterFile -b -L -t 10
```

This command is identical to the previous one for measuring random DN bind and search rate, except it uses a UID file (see "DN and UID File Formats," on page 95). Here, in each repeated search sequence, the command will first select a random UID from the UID file (`-B` *UIDfile* `-x`), bind anonymously (`-D "" -w ""`) to find the DN of the corresponding entry, and then bind as this DN using the UID for authentication before performing the search.

You may need to set the `-D` and `-w` options for valid authentication if your directory does not allow anonymous binding.

Measuring Compare Operations

The following examples will measure the performance of compare operations. The `-s ""` and `-f ""` options have no effect but are required on the command line.

Compare Rate

```
$ rsearch -h hostname -p port -s "" -f "" \
          -D "bindDN" -w password -B DNfile -c -t 2 -j 5 -v
```

This command will create two threads (`-t 2`), each of which will bind once using the given *bindDN* and *password* (use `-D "" -w ""` for anonymous binding), repeatedly perform a compare operation (`-c`) on the `uid` attribute of random DNs from the DN file, and display verbose results every 5 seconds (`-j 5 -v`).

Bind and Compare Rate

```
$ rsearch -h hostname -p port -s "" -f "" -D "" -w "" \
          -B DNfile -x -c -b -L -t 2 -j 5 -v
```

In this form, the `-D ""` and `-w ""` options have no effect but are required on the command line.

This command will create two threads (`-t 2`), each of which will bind before every compare operation (`-c -b -L`) and unbind afterwards, while displaying verbose results every 5 seconds (`-j 5 -v`). Binding will use a random DN from the DN file (`-B` *DNfile* `-x`) with the DN's corresponding UID as the password. Another DN will be randomly selected from the same file and used the target of the compare operation based again on its UID. This implies that directory entries for DNs appearing in the *DNfile* must have the same UID and password.

Measuring Modify Operations

The following examples will measure the performance of modify operations. The rsearch tool performs modify operation on the `telephonenumber` and `description` attributes. These examples assume that the `telephonenumber` attribute is indexed in your test directory and that the `description` attribute is not.

In these examples, the `-s ""` and `-f ""` options have no effect but are required on the command line. Also, adding the -x option will use the same DN file for both binding and modification. In this case however, all DNs in your DN file must have modification rights to all entries corresponding to those DNs.

Indexed Attribute Modify Rate

```
$ rsearch -h hostname -p port -s "" \
          -D "cn=Directory Manager" -w password -B DNfile -M

$ rsearch -h hostname -p port -s "" \
          -D "cn=Directory Manager" -w password -B DNfile -M -b
```

These commands will bind as the root DN (`-D "cn=Directory Manager" -w password`) and repeatedly perform a modify operation on the `telephonenumber` attribute of randomly chosen entries from the DN file (`-B` *DNfile* `-M`). While this example scenario assumes that the `telephonenumber` attribute is indexed, this command does not verify this and will function normally even if it is not.

The first form of this command will bind only once and measure the average rate of modify operations. The second form with the `-b` option will rebind at every modify operation to measure the average time for the bind and modify operation sequence.

Non-Indexed Attribute Modify Rate

```
$ rsearch -h hostname -p port -s "" \
          -D "cn=Directory Manager" -w password -B DNfile -m

$ rsearch -h hostname -p port -s "" \
          -D "cn=Directory Manager" -w password -B DNfile -m -b
```

These commands will bind as the root DN (`-D "cn=Directory Manager" -w password`) and repeatedly perform a modify operation on the `description` attribute of randomly chosen entries from the DN file (`-B` *DNfile* `-m`). While this example scenario assumes that the `description` attribute is not indexed, this command does not verify this and will function normally even if it is.

The first form of this command will bind only once and measure the average rate of modify operations. The second form with the `-b` option will rebind at every modify operation to measure the average time for the bind and modify operation sequence.

Measuring Delete Operations

The following examples will measure the performance of delete operations in your directory. The `-s ""` and `-f ""` options have no effect but are required on the command line.

Delete Rate

```
$ rsearch -h hostname -p port -s "" -f ""
            -D "cn=Directory Manager" -w password -B DNfile -d -T 7
```

This command will bind once as the root DN (`-D "cn=Directory Manager" -w` *password*) and repeatedly delete entries whose DN was randomly chosen from the DN file (`-B` *DNfile* `-d`).

Because DNs are randomly chosen from the DN file, errors will occur when the tool tries to delete entries that have already been chosen and deleted. To minimize these errors, set the time limit option (`-T 7`) and use a DN file with a large number of DNs.

Bind and Delete Rate

```
$ rsearch -h hostname -p port -s "" -f "" -D "cn=Directory Manager"
            -w password -B DNfile -d -b -L -T 7
```

This command will repeatedly perform delete operations on entries randomly chosen from the DN file (`-B` *DNfile* `-d`), while rebinding every time (`-b -L`) as the root DN (`-D "cn=Directory Manager" -w` *password*).

NOTE Do not use the `-x` option when performing bind and delete operations: this will cause `rsearch` to attempt to bind with DNs that may have already been deleted.

Chapter 9

searchrate

The searchrate tool measures the performance of search operations in an LDAP v3 directory. It is similar to the rsearch tool described previously, except that it performs searches only.

As with all measures of performance, results depend upon many factors, such as the options and parameter values given, directory configuration, machine load, and network traffic, and should be analyzed accordingly. In order to obtain accurate performance measurements, both client and server machines should be dedicated to the performance test and properly tuned (see Chapter 7, “idsktune”).

This chapter contains the following sections:

- Command Usage
- Sample Output
- Command-Line Examples

Command Usage

Using multiple threads, the searchrate tool simulates a search load on a directory server. Each thread performs LDAP bind and search operations repeatedly as often as possible, and the tool displays average results at regular intervals. The command-line options let you configure the binding sequence and the scope of the searches. The searchrate tool has the following built-in defaults:

- All operations use the LDAP v3 protocol. The tool cannot be used to test directories that only support LDAP v2.
- The tool uses simple or anonymous binding. No secure binding is possible.
- Referrals are never followed.

- The time and size limits for search results are not modifiable. The default time limit for a synchronous search is 10 seconds. All other default values are those defined by the directory server.

In general, when the `searchrate` tool encounters an error, it displays a message and continues running. It will attempt to bind again or search again indefinitely, even after encountering an error.

Syntax

The `searchrate` command has the following syntax:

`searchrate -b "`*baseDN*`" -f "`*filter*`" [` *options* `]`

Where:

- *baseDN* is the suffix, usually in double quotes (`""`) for the shell, which represents the subtree to be searched.
- *filter* is an RFC 2254-compliant LDAP search filter, usually in double quotes (`""`) for the shell (see “LDAP Search Filters” in Appendix B of the *iPlanet Directory Server Administrator’s Guide*).
- *options* are the command-line options and their parameters described in the next section.

Both the base DN and filter strings may use the following syntax (see “Random Searches,” on page 107):

  - ❍ `%d` to include random numbers up to the value given by the `-r` option.
  - ❍ `%s` to include a random string from the file given by the `-i` option.

Running the `searchrate` command without any options or parameters will display the usage help text that briefly describes all options.

Options

The `searchrate` options and parameters are described in the following table.

Table 9-1 Command-Line Options for the `searchrate` Tool

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. The default is `localhost`.

Table 9-1 Command-Line Options for the `searchrate` Tool *(Continued)*

Option	Parameter	Purpose
`-p`	*port*	Specify the port number when accessing the directory server host. The default is `389`.
`-D`	*bindDN*	Specify a bind DN for accessing the directory, usually in double quotes (`""`) for the shell. Depending on its access permissions, the bind DN may influence authentication and search performance. If the bind DN and its password are omitted, the tool will perform search operations without binding (as permitted by LDAP v3). For anonymous binding, you must explicitly specify an empty bind DN and password (`-D "" -w ""`).
`-w`	*password*	Specify the password for the bind DN.
`-b`	*baseDN*	Specify the base DN for the search operations, usually in double quotes (`""`) for the shell. See "Random Searches," on page 107 on how to use `%s` and `%d` placeholders for including random numbers and strings with the `-r` and `-i` options, respectively.
`-s`	*scope*	Specify the scope of a search with one of the following values: `base` - For searching only the base entry. `one` - For searching one level below the base entry. `sub` - For searching all levels below the entry. This is the default.
`-f`	*filter*	Specify the filter for all search operations. See "Random Searches," on page 107 on how to use `%s` and `%d` placeholders for including random numbers and strings with the `-r` and `-i` options, respectively.
`-g`	*seconds*	Specify the timelimit in seconds for each search operation. The default is 10 seconds. You may need to set the limit higher if you specify a broad search, such as `(cn=*)` over a large directory.
`-i`	*inputFile*	Give the name of the file containing strings that will be randomly substituted into `%s` placeholders in the base DN and filter. Each line of the input file is treated as a separate string. See "Random Searches," on page 107 for more information.
`-r`	*maxRand*	Give the maximum range for random numbers to be substituted into `%d` placeholders in the base DN and filter. You may specify this option twice: the first random number will be in the range `[0,` *maxRand1*`-1]`, the second will be in the range `[1,` *maxRand2*`]`.
`-A`	*attribute*	Specify an attribute to be retrieved during search operations. You may use this option any number of times on the command line: only the specified attributes will be retrieved. When omitted, all searches will return all attributes for all matching entries. Whether or not this attribute is used, no attribute values are ever displayed. However, retrieving different attributes may influence search performance.

Table 9-1 Command-Line Options for the `searchrate` Tool *(Continued)*

Option	Parameter	Purpose
`-k`		Keep connections open between searches. With this option, the `searchrate` tool will measure only the execution time of the bind and search operations. When this option is omitted, the initialization and freeing of the connection are also measured as part of each search sequence.
`-K`		Keep connections and binds open between searches. With this option, the `searchrate` tool will measure only the execution time of the search operation. When this option is omitted, the initialization, binding, unbinding, and freeing of the connection is also measured as part of each search sequence.
`-u`		When used with the `-D -w` options, specify that the tool should not unbind from the sever, just close the socket for the connection. This option has no effect when either `-k` or `-K` options are specified.
`-a`		Specify asynchronous search mode. With this option, the `searchrate` tool will not wait for search results before starting the next search. As a result, performance measurements will reflect `searchrate` execution speed, not directory server search speed. This option cannot be used when the `-D` and `-w` options are given.
`-t`	*threads*	Specify the number of threads that `searchrate` will run in parallel. The output displays the average performance of all threads combined. The default is a single thread.
`-j`	*seconds*	Specify the measurement and display interval, in seconds; the default is `5`. `searchrate` repeats the search sequence as many times as possible during the interval and prints results after each interval elapses.
`-m`	*searchOps*	Specify the maximum, cumulated number of search operations for each thread to perform. When this option is not specified, all threads will repeat the search sequence indefinitely.
`-q`		Quiet output mode: the measurements for each interval will not be displayed.

Random Searches

One concern for accurate performance measurements is to simulate real usage conditions and reduce any artifacts due to the repetitive nature of the tests. For example, when the same entry is retrieved for every search, it will be cached by the directory server and returned without performing a full search. Under normal usage conditions, only a certain percentage of searches will be resolved quickly through the entry cache, so the test results will be skewed.

To simulate real usage conditions, the `searchrate` tool includes a mechanism for randomizing both the base DN and the search filter used for each search. You can include either randomly generated numbers or random strings from an input file. To do this, specify the following placeholders in the base DN, in the filter parameters, or in both on the command line:

- The first occurrence of `%d` will be replaced by a random number in the range `[0,` *maxRand1*`-1]`, where *maxRand1* is given by the first occurrence of the `-r` option on the command line.
- The second occurrence of `%d` will be replaced by a random number in the range `[1,` *maxRand2*`]`, where *maxRand2* is given by the second occurrence of the `-r` option on the command line.
- The `%s` placeholder will be replaced by a random string from the *inputFile* given by the `-i` option. Each line of this file is treated as a complete string.

The tool applies the following rules for substitutions. An offending command line will return a usage error:

- You must specify at least as many `-r` options as `%d` placeholders you use.
- Placeholder substitution will occur only in the base DN and search filter parameters. To use the literal strings `"%d"` and `"%s"` within these parameters, you must use `"%%d"` and `"%%s"`, respectively.
- Within the same parameter, you may use only one type of placeholder. However, each parameter may use a different type.
- When both parameters specify the same type of placeholder, the same value or values will be substituted in both for a given search.

The input file is a plain text file that contains the strings for substitution. Each line, including any whitespace, is taken to be one string. File contents should be adapted to the intended substitutions: either in the base DN, in the search filter, or in both. For example, the following input file could be used with the option `-f "sn%s"` for performing different searches on surnames:

```
=C*
=Jones
=Smith
>=Jones
<=Jones
~=Turner
```

The size and contents of the file should be adapted to the directory that you are testing. Generally, the type and number of different searches that can be chosen randomly from the input file should resemble the expected usage of your directory.

Sample Output

```
$ searchrate -h hostname -b "dc=siroe,dc=com" -f "cn=a*" -t 4
```

```
Avg r=  74.75/thr ( 59.80/sec), total=   299
Avg r=  76.00/thr ( 60.80/sec), total=   304
Avg r=  74.50/thr ( 59.60/sec), total=   298
Avg r=  56.00/thr ( 44.80/sec), total=   224
Avg r=  73.50/thr ( 58.80/sec), total=   294
Avg r=  75.25/thr ( 60.20/sec), total=   301
^C
```

When running, the `searchrate` tool displays one line of measurements every interval (5 seconds by default). All data on an output line concerns only the elapsed interval. Use the `-j` *seconds* option to specify a different interval length. Reading an output line backwards, it shows:

- The total number of search operations completed by all threads during the full interval.
- The rate in parentheses is the average number of searches per second for all threads (the total divided by the number of seconds in the interval).
- The given `Avg` is the average number of operations per thread during the interval (the total divided by the number of threads).

Command-Line Examples

The examples in this section will measure bind and search performance in your directory server in various scenarios. Results will be meaningful only if factors such as machine load and network traffic remain constant during and between tests.

In these examples, you will need to provide the base DN and filter strings for valid entries in your test directory. You will need to adapt these examples to your environment:

- The *hostname* and *port* placeholders should be replaced with the hostname and port number of your directory. The *baseDN* should represent the root of the directory or subtree you wish to test.
- For realistic results, thread numbers and time limit options should be scaled according to your directory's expected load.
- You will also need to provide input files that correspond to the contents of your directory (see "Random Searches," on page 107).

Input files may also contain different kinds of filter strings that may give different performance results:

- Strings containing wildcards will give measurements that reflect substring search speed.
- Strings without wildcards will give measurements that reflect the search speed for exact matches.
- Filter strings may also include operators other than equality, such as ranges that are greater than (>=) or less than (<=) a given value, or approximate spelling searches (~=).

Simple Search

```
$ searchrate -h hostname -p port -b "dc=Siroe,dc=com" \
             -f "cn=*john*" -t 3 -j 60
```

This command will launch 3 threads (`-t 3`), each of which will repeatedly open a connection but not bind, search for entries under the `"dc=Siroe,dc=com"` suffix that contain the `"cn=*john*"` substring, and close the connection (no `-D -w -k -K` options). The tool will display combined results for all threads at one minute intervals (`-j 60`).

Because every search uses the same filter and returns the same entries, entry caching in the directory server will influence results. In all of the following examples, the use of an input file containing filter strings helps provide more realistic measurements.

Open, Bind, and Search Rate

```
$ searchrate -h hostname -p port -b "dc=Siroe,dc=com" \
             -D "bindDN" -w password -f "sn%s" -i inputFile
```

This command will use a single thread to repeatedly open a connection, bind with the given credentials (`-D "`*bindDN*`" -w` *password*), perform a search under the `"dc=Siroe,dc=com"` suffix and then unbind. The unbinding operation also closes the connection. During each iteration, the tool will search for a different, random surname taken from the input file (`-f "sn=%s" -i` *inputFile*).

```
$ searchrate -h hostname -p port -b "dc=Siroe,dc=com" \
             -D "bindDN" -w password -u -f "sn%s" -i inputFile
```

This command is similar to the previous one, except that at each iteration, the connection and binding will simply be dropped by closing the socket (`-u`). This behavior is allowed by the LDAP protocol, and this test verifies that connections that are not unbound are properly handled by the directory server.

Both of the previous commands may also use anonymous binding (`-D "" -w ""`) that may give different performance results due to handling anonymous access permissions.

Bind and Search Rate

```
$ searchrate -h hostname -p port -b "dc=Siroe,dc=com" \
             -D "bindDN" -w password -k -f "sn%s" -i inputFile
```

This command will use a single thread to keep a connection open (`-k`) to repeatedly bind with the given credentials and perform a search under the `"dc=Siroe,dc=com"` suffix. The LDAP protocol allows clients to bind multiple times without unbinding, and this test measures performance in this situation.

During each iteration, the tool will search for a different, random surname taken from the input file (`-f "sn=%s" -i` *inputFile*). You may also use anonymous binding (`-D "" -w ""`) to test the performance results using anonymous access.

Search Rate Alone

```
$ searchrate -h hostname -p port -b "dc=Siroe,dc=com" \
             -D "bindDN" -w password -K -f "sn%s" -i inputFile
```

This command will use a single thread to keep the connection and the bind open (-K) and repeatedly perform a search under the "dc=Siroe,dc=com" suffix. During each iteration, the tool will search for a different, random surname taken from the input file (-f "sn=%s" -i *inputFile*). This will isolate the performance measurements of the search operation alone. You may also use anonymous binding (-D "" -w "") to test the performance results using anonymous access.

```
$ searchrate -h hostname -p port -b "dc=Siroe,dc=com" \
             -k -f "sn%s" -i inputFile
```

In this command, the single thread will only open a connection without binding (-k but no -D -w) and repeatedly perform search operations. Unbound search operations are allowed by the LDAP v3 protocol, and this test isolates the performance of the search operation alone in this situation.

Chapter 10

modrate

The `modrate` tool measures the performance of modify operations in an LDAP v3 directory. It is similar to the `rsearch` functionality described previously, except that it performs modifications on random user-defined attributes.

As with all measures of performance, results depend upon many factors, such as the options and parameter values given, directory configuration, machine load, and network traffic, and should be analyzed accordingly. In order to obtain accurate performance measurements, both client and server machines should be dedicated to the performance test and properly tuned (see Chapter 7, “idsktune”).

This chapter contains the following sections:

- Command Usage
- Sample Output
- Command-Line Examples

Command Usage

Using multiple threads, the `modrate` tool repeatedly performs modify operations on a directory server. Threads may be configured to open connections and perform LDAP bind operations with every modification. The command-line options let you specify the target entries and attributes to be modified. The `modrate` tool has the following built-in defaults:

- All operations use the LDAP v3 protocol. The tool cannot be used to test directories that only support LDAP v2.
- The tool uses simple or anonymous binding. No secure binding is possible.
- Referrals are never followed.

- The time limit for operations is not modifiable. The default time limit is that defined by the directory server.

The tool displays performance results at regular intervals. In general, when the `modrate` tool encounters an error, it displays a message and continues running. It will attempt to bind again or modify again indefinitely, even after encountering an error.

Syntax

The `modrate` command has the following syntax:

```
modrate -D "bindDN" -w password -b "baseDN" [ options ] \
        -M "attribute:length:charSet" ...
```

Where:

- *bindDN* and *password* are bind credentials with write permission to the target entry or enties. The bind DN is usually in double quotes ("") for the shell.
- *baseDN* is the DN of the entry to be modified, usually in double quotes ("") for the shell. The target entry should support the attributes given by the `-M` option, either for modification or for addition when not already present. The base DN may use either of the following placeholders (see "Random Modifications," on page 116):
  - `%d` to include random numbers up to the value given by the `-r` option.
  - `%s` to include a random string from the file given with the `-i` option.
- *options* are the command-line options and their parameters described in the next section.
- The `-M` parameter gives the information needed to generate random attribute values. There can be any number of `-M` parameters on the command line.
  - *attribute* is the name of an existing attribute of the target entry or entries given by the base DN. If this attribute does not exist on the target entry, it will be added and be subject to schema checking if it is enabled in the directory.
  - *length* is an integer giving the desired number of characters in random attribute string values.
  - *charSet* specifies a set of individual ASCII charaters (*c*) and ranges of characters, according to the following syntax:

    `(c*([c-c])*)*` for example: `[A-Z][a-z][0-9]`

> For each modify operation, a `modrate` thread will randomly choose one of the *attribute* names and generate a new string value of the given *length* by choosing each character randomly from the given *charSet*.

Running the `modrate` command without any options or parameters will display the usage help text that briefly describes all options.

Options

The `modrate` options and parameters are described in the following table.

Table 10-1 Command-Line Options for the `modrate` Tool

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. The default is `localhost`.
`-p`	*port*	Specify the port number when accessing the directory server host. The default is `389`.
`-D`	*bindDN*	Specify a bind DN for accessing the directory, usually in double quotes (`""`) for the shell. Depending on its write permissions, the bind DN may influence authentication and modify performance.
`-w`	*password*	Specify the password for the bind DN.
`-b`	*baseDN*	Specify the base DN of the target entry, usually in double quotes (`""`) for the shell. See “Random Modifications,” on page 116 on how to include `%s` and `%d` placeholders for random numbers and strings with the `-r` and `-i` options, respectively.
`-M`	*modString*	Specify the name of an attribute to modify and how to randomly generate a new value for it. The *modString* format (*attribute*:*length*:*charSet*) is described in “Syntax,” on page 114.
`-i`	*inputFile*	Give the name of the file containing strings that will be randomly substituted into `%s` placeholders in the base DN. Each line of the input file will be treated as a separate string. See “Random Modifications,” on page 116 for more information. This option is incompatible with the `-r` option.
`-r`	*maxRand*	Give the maximum range for random numbers to be substituted into `%d` placeholders in the base DN. You may specify this option twice: the first random number will be in the range `[0,` *maxRand1*`-1]`, the second will be in the range `[1,` *maxRand2*`]`. This option is incompatible with the `-i` option.

Table 10-1 Command-Line Options for the `modrate` Tool *(Continued)*

Option	Parameter	Purpose
`-k`		Keep connections open between modify operations. With this option, the `modrate` tool will measure only the execution time of bind and modify operations. When this option is omitted, the initialization and freeing of the connection is also measured as part of each modification sequence.
`-K`		Keep connections and binds open between modify operations. With this option, the `modrate` tool will measure only the execution time of modify operations. When this option is omitted, the initialization, binding, unbinding, and freeing of the connection is also measured as part of each sequence.
`-u`		Specify that the tool should not unbind from the server and just close the socket for the connection. This option has no effect when either `-k` or `-K` options are specified.
`-a`		Specify asynchronous modification mode. This option is currently not working.
`-t`	*threads*	Specify the number of threads that `modrate` will run in parallel. The output displays the average performance of all threads combined. The default is a single thread.
`-j`	*seconds*	Specify the measurement and display interval, in seconds; the default is `5`. `modrate` repeats the modify operation sequence as many times as possible during the interval and prints results after each interval elapses.
`-m`	*modifyops*	Specify the maximum, cumulated number of modify operations for each thread to perform. When this option is not specified, all threads will repeat the modify sequence indefinitely.
`-q`		Quiet output mode: the measurements for each interval will not be displayed.

Random Modifications

To simulate real usage conditions and reduce any artifacts due to the repetitive nature of the tests, the `modrate` tool provides a mechanism for choosing a random target entry. You can include either randomly generated numbers or random strings from an input file. To do this, specify the following placeholders in the base DN parameter on the command line:

- The first occurrence of `%d` will be replaced by a random number in the range `[0,` *maxRand1*`-1]`, where *maxRand1* is given by the first occurrence of the `-r` option on the command line.
- The second occurrence of `%d` will be replaced by a random number in the range `[1,` *maxRand2*`]`, where *maxRand2* is given by the second occurrence of the `-r` option on the command line.
- The `%s` placeholder will be replaced by a random string from an *inputFile* given by the `-i` option. Each line of this file is treated as a complete string to insert.

The tool applies the following rules for substitutions. An offending command line will return a usage error:

- You must specify at least as many `-r` options as `%d` placeholders you use.
- Placeholder substitution will occur only in the base DN parameters. To use the literal strings `"%d"` and `"%s"` within this parameter, you must use `"%%d"` and `"%%s"`, respectively.
- You may use only one type of placeholder, either decimal number or string.

The input file is a plain text file that contains the strings for substitution. Each line, including any whitespace, is taken to be one string. Depending on how the placeholder is used, the file may contain full or partial DNs. For example, the `-b "uid=%s,ou=people,dc=Siroe,dc=com"` option could be used with the following input file:

```
bjensen
bjense2
mtalbot
kcarter
svaughan
pshelton
```

The size and contents of the file should be adapted to the directory that you are testing. Generally, the number of possible target entries should be as large as possible to ensure performance is being measured across the whole directory.

Sample Output

```
$ modrate -h hostname -D "cn=directory manager" -w password \
          -b "uid=bjensen,ou=people,dc=Siroe,dc=com" \
          -M telephonenumber:10:[0-9]
```

```
Avg r=  37.00/thr (  7.40/sec), total=    37
Avg r=  36.00/thr (  7.20/sec), total=    36
Avg r=  29.00/thr (  5.80/sec), total=    29
Avg r=  35.00/thr (  7.00/sec), total=    35
Avg r=  38.00/thr (  7.60/sec), total=    38
Avg r=  39.00/thr (  7.80/sec), total=    39
^C
```

When running, the `modrate` tool displays one line of measurements every interval (5 seconds by default). All data on an output line concerns only the elapsed interval. Use the `-j` *seconds* option to specify a different interval length. Reading an output line backwards, it shows:

- The total number of modify operations completed by all threads during the full interval.
- The rate in parentheses is the average number of modify operations per second for all threads (the total divided by the number of seconds in the interval).
- The given `Avg` is the average number of operations per thread during the interval (the total divided by the number of threads).

Command-Line Examples

The examples in this section measure bind and modify performance in your directory server in various scenarios. Results are meaningful only if factors such as machine load and network traffic remain constant during and between tests.

In these examples, you should provide DNs for modifiable entries in your test directory. You will need to adapt these examples to your environment:

- The *hostname* and *port* placeholders should be replaced with the hostname and port number of your directory.
- For realistic results, thread numbers should be scaled according to your directory’s expected load.
- You should also specify modifiable attributes in the target entries with character sets that conform to your directory’s schema.

Open, Bind and Modify Rate

```
$ modrate -h hostname -p port -D "bindDN" -w password \
          -b "uid=bjensen,ou=people,dc=Siroe,dc=com" \
          -M telephonenumber:10:[0-9] -t 3 -j 60
```

This command will launch 3 threads (`-t 3`), each of which will repeatedly open a connection, bind with the given credentials (`-D "`*bindDN*`" -w` *password*), modify the `telephonenumber` attribute of the `bjensen` entry, unbind, and close the connection (no `-u -k -K` options). The new attribute value for each modification is a sequence of 10 random digits (`-M telephonenumber:10:[0-9]`) that simulate a telephone number. The tool will display combined results for all threads at one minute intervals (`-j 60`).

Adding the `-u` option to this command line will test whether the directory server handles clients that don't unbind before disconnecting.

Because every modify operation applies to the same entry, entry caching in the directory server will influence results. In the following examples, the use of an input file containing UID strings helps provide more realistic measurements (see "Random Modifications," on page 116).

Bind and Modify Rate

```
$ modrate -h hostname -p port -D "bindDN" -w password \
          -b "uid=%s,ou=people,dc=Siroe,dc=com" -i inputFile \
          -k -M telephonenumber:10:[0-9]
```

This command will use a single thread to keep a connection open (`-k`) to repeatedly bind with the given credentials and perform a modify operation on an entry whose UID is randomly chosen from the *inputFile*. The LDAP protocol allows clients to bind multiple times without unbinding, and this test measures performance in this situation.

Modify Rate Alone

```
$ modrate -h hostname -p port -D "bindDN" -w password \
          -b "uid=%s,ou=people,dc=Siroe,dc=com" -i inputFile \
          -K -M telephonenumber:10:[0-9]
```

This command will use a single thread to keep the connection and the bind open (`-K`) and repeatedly perform a modify operation on an entry whose UID is randomly chosen from the *inputFile* at each iteration. This will isolate the performance measurements of the modify operation alone.

Chapter 11

authrate

The authrate tool measures the possible rate of authentication to an LDAP v3 directory. It is similar to the rsearch functionality described previously, providing a mechanism for using random bind DN and password credentials.

As with all measures of performance, results depend upon many factors, such as the options and parameter values given, directory configuration, machine load, and network traffic, and should be analyzed accordingly. In order to obtain accurate performance measurements, both client and server machines should be dedicated to the performance test and properly tuned (see Chapter 7, “idsktune”).

This chapter contains the following sections:

- Command Usage
- Sample Output
- Command-Line Examples

Command Usage

Using multiple threads, the authrate tool repeatedly initializes a connection and binds to a directory server, without performing any other operation. Threads may be configured to keep open connections and perform LDAP binds repeatedly. The command-line options let you specify the bind credentials. The modrate tool has the following built-in defaults:

- All operations use the LDAP v3 protocol. The tool cannot be used to test directories that only support LDAP v2.
- The tool uses simple or anonymous binding. No secure binding is possible.

The tool displays performance results at regular intervals. In general, when the `authrate` tool encounters an error, it displays a message and continues running. It will attempt to bind again indefinitely, even after encountering an error.

Syntax

The `authrate` command has the following syntax:

```
authrate -D "bindDN" -w password [ options ]
```

Where:

- *bindDN* and *password* are the bind credentials, with the bind DN is usually in double quotes (`""`) for the shell. The bind DN and password may use the `%d` placeholder to include random numbers (see “Random Authentication,” on page 123).
- *options* are the command-line options and their parameters described in the next section.

The `authrate -H` command will display the usage help text that briefly describes all options.

Options

The `authrate` options and parameters are described in the following table.

Table 11-1 Command-Line Options for the `authrate` Tool

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. The default is `localhost`.
`-p`	*port*	Specify the port number when accessing the directory server host. The default is `389`.
`-D`	*bindDN*	Specify a bind DN for accessing the directory, usually in double quotes (`""`) for the shell. See “Random Authentication,” on page 123 on how to include up to two `%d` placeholders for random numbers using the `-r` option.
`-w`	*password*	Specify the password for the bind DN. The password may also contain `%d` placeholders that will use the same random numbers as the bind DN substitutions. See “Random Authentication,” on page 123 for details.

Table 11-1 Command-Line Options for the `authrate` Tool *(Continued)*

Option	Parameter	Purpose
`-r`	*maxRand*	Give the maximum range for random numbers to be substituted into `%d` placeholders in the base DN and password. You may specify this option twice: the first random number will be in the range `[0,` *maxRand1*`-1]`, the second will be in the range `[1,` *maxRand2*`]`.
`-k`		Keep connections open when performing binds. With this option, the `authrate` tool will measure only the execution time of the bind operation. When this option is omitted, the initialization and freeing of the connection is also measured as part of each authentication sequence.
`-u`		Specify that the tool should not unbind from the server and just close the socket for the connection. This option has no effect when the `-k` option is specified.
`-t`	*threads*	Specify the number of threads that `authrate` will run in parallel. The output displays the average performance of all threads combined. The default is a single thread.
`-j`	*seconds*	Specify the measurement and display interval, in seconds; the default is `5`. `authrate` repeats the authentication sequence as many times as possible during the interval and prints results after each interval elapses.
`-m`	*bindOps*	Specify the maximum number of bind operations for each thread to perform. When this option is not specified, all threads will repeat the authentication sequence indefinitely.
`-q`		Quiet output mode: the measurements for each interval will not be displayed.

Random Authentication

To simulate real usage conditions and reduce any artifacts due to the repetitive nature of the tests, the `authrate` tool provides a mechanism for using a random bind DN. You can include randomly generated numbers by specifying the following placeholders:

- In the bind DN, the first and second occurrences of `%d` will be replaced by a random number in the ranges `[0,` *maxRand1*`-1]` and `[1,` *maxRand2*`]`, respectively, where *maxRand1* and *maxRand2* are given by the first and second occurrences of the `-r` option on the command line. The bind DN parameter may have no more than two `%d` placeholders.

- In the password parameter, all occurrences of `%d` will be replaced by a random number in the ranges `[0,` *maxRand1*`-1]`, where *maxRand1* is given by the first occurrence of the `-r` option on the command line. The password parameter may have up to 8 placeholders, to generate a password with enough characters when the random substitution is a single digit.

To use the random authentication, your test directory must contain entries written with these substitutions rules in mind. Because the same random number will be substituted into both bind DN and password, the mechanism generates matched DN and password pairs. For example, the following entries will work with the given command line:

Table 11-2 Random Authentication

Entries for using random authentication	`dn: cn=test0,dc=siroe,dc=com`
	`password: auth00`
	`dn: cn=test1,dc=siroe,dc=com`
	`password: auth11`
	`dn: cn=test2,dc=siroe,dc=com`
	`password: auth22`
	`...`
	`dn: cn=test10,dc=siroe,dc=com`
	`password: auth1010`
	`...`
	`dn: cn=test99,dc=siroe,dc=com`
	`password: auth9999`

Corresponding command line	**`authrate -D "cn=test%d,dc=siroe,dc=com" \`**
	`-w "auth%d%d" -r 100`

Sample Output

```
$ authrate -h hostname -p port \
           -D "cn=test%d,dc=siroe,dc=com" -w "auth%d%d" -r 100

Avg r= 754.00/thr (150.80/sec), total=   754
Avg r= 774.00/thr (154.80/sec), total=   774
Avg r= 829.00/thr (165.80/sec), total=   829
Avg r= 825.00/thr (165.00/sec), total=   825
Avg r= 836.00/thr (167.20/sec), total=   836
Avg r= 837.00/thr (167.40/sec), total=   837
^C
```

When running, the `authrate` tool displays one line of measurements every interval (5 seconds by default). All data on an output line concerns only the elapsed interval. Use the `-j` *seconds* option to specify a different interval length. Reading an output line backwards, it shows:

- The total number of authentications completed by all threads during the full interval.
- The rate in parentheses is the average number of authentications per second for all threads (the total divided by the number of seconds in the interval).
- The given `Avg` is the average number of authentications per thread during the interval (the total divided by the number of threads).

Command-Line Examples

The examples in this section will measure authentication performance in your directory server in various scenarios. Results will be meaningful only if factors such as machine load and network traffic remain constant during and between tests.

These examples suppose the contents of your test directory are configured as explained in “Random Authentication,” on page 123. You will need to adapt other parameters to your environment:

- The *hostname* and *port* placeholders should be replaced with the hostname and port number of your directory.
- For realistic results, thread numbers should be scaled according to your directory’s expected load.

Open and Bind Rate

```
$ authrate -h hostname -p port -t 3 -j 60 \
           -D "cn=test%d,dc=siroe,dc=com" -w "auth%d%d" -r 100
```

This command will launch 3 threads (`-t 3`), each of which will repeatedly open a connection, bind with randomly generated credentials (`-D "cn=test%d,dc=siroe,dc=com" -w "auth%d%d" -r 100`), unbind, and close the connection (no `-u -k` options). The tool will display combined results for all threads at one minute intervals (`-j 60`).

Adding the `-u` option to this command line will test whether the directory server handles clients that don’t unbind before disconnecting.

Bind Rate Alone

```
$ authrate -h hostname -p port -k \
           -D "cn=test%d,dc=siroe,dc=com" -w "auth%d%d" -r 100
```

This command will use a single thread to keep a connection open (`-k`) to repeatedly bind with randomly generated credentials (`-D "cn=test%d,dc=siroe,dc=com" -w "auth%d%d" -r 100`). The LDAP protocol allows clients to bind multiple times without unbinding, and this test measures performance in this situation.

Chapter 12

infadd

The `infadd` (infinite add) tool measures the performance of add operations in an LDAP v3 directory. It generates entries containing random attribute values and adds them to the directory under a given suffix. It performs operations continuously and computes an average operation rate at regular intervals.

As with all measures of performance, results depend upon many factors, such as the options and parameter values given, directory configuration, machine load, and network traffic, and should be analyzed accordingly. In order to obtain accurate performance measurements, both client and server machines should be dedicated to the performance test and properly tuned (see Chapter 7, “idsktune”).

This chapter contains the following sections:

- Command Usage
- Command-Line Examples

Command Usage

Using multiple threads, the `infadd` tool binds to a directory server and repeatedly performs LDAP add operations. All entries are added to the same subtree, one level below the *suffix* given on the command line. New entries belong to the `inetOrgPerson` object class and have the following attributes:

```
dn: cn= givenname sn UID,suffix
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: givenname sn UID
givenname: givenname
sn: sn
```

```
uid: UID
mail: givennameUID@siroe.com
telephonenumber: telephonenumber
userpassword: UID
audio: binaryData
```

The *givenname* and *sn* are randomly chosen either from data files containing names or from randomly generated strings. The following data files are provided in the `data` subdirectory of the iPlanet Directory Server Resource Kit:

- `dbgen-FamilyNames` contains over 13400 plausible surnames.
- `dbgen-GivenNames` contains over 8600 plausible first names.

The *UID* is a sequential numbering of new entries (see the `-I` option), and the *telephonenumber* is a random, US-format telephone number. The binary `audio` attribute is optional and can be used to test the addition of large entries (see the `-z` option).

The tool displays performance results at regular intervals. Performance of add operations is highly dependent upon whether or not the directory performs schema checking. Be sure your directory is configured so that the test scenario most closely resembles its actual usage.

Syntax

The `infadd` command has the following syntax:

```
infadd -s "suffix" -u "bindDN" -w password [ options ]
```

Where:

- *suffix* is the base DN under which all entries will be added.
- *bindDN* and *password* are the bind credentials, with the bind DN is usually in double quotes (`""`) for the shell.
- *options* are the command-line options and their parameters described in the next section.

Running the `infadd` command without any options or parameters will display the usage help text that briefly describes all options.

Options

The `infadd` options and parameters are described in the following table.

Table 12-1 Command-Line Options for the `authrate` Tool

Option	Parameter	Purpose
`-h`	*hostname*	Specify the hostname of the directory server. The default is `localhost`.
`-p`	*port*	Specify the port number when accessing the directory server host. The default is `389`.
`-u`	*bindDN*	Specify a bind DN for accessing the directory, usually in double quotes (`""`) for the shell. The bind DN should have write permission in the subtree given by the `-s` *suffix* parameter.
`-w`	*password*	Specify the password for the bind DN.
`-s`	*suffix*	Specify the suffix to use for all new entries, usually in double quotes (`""`) for the shell. This is effectively the base DN under which all entries will be added.
`-d`		Set the `TCP_NODELAY` mode on the TCP socket.
`-t`	*threads*	Specify the number of threads that `infadd` will run in parallel. Use the `-v` option for verbose output including measurements from each thread. The default is a single thread.
`-i`	*millisecs*	Specify the measurement and display interval, in milliseconds; the default is `10000`. `infadd` creates as many new entries as possible during the interval and prints results after each interval elapses.
`-l`	*addOps*	Specify the approximate number of total add operations for the tool to perform. The tool will stop after the measurement interval where the total number of operations for all threads exceeds this parameter. When this option is not specified, all threads will continue adding entries indefinitely.
`-q`		Quiet output mode: the measurements for each interval will not be displayed.
`-v`		Verbose output at every interval: gives the measurements from each thread, including the minimum and maximum operation times observed, as well as the average over all threads.
`-I`	*startID*	For guaranteeing uniqueness of DNs, the `infadd` tool generates a sequential ID number, beginning with *startID*. This ID number is appended to the `cn` attribute and also used as the `uid` attribute of new entries. When this option is omitted, ID numbers begin at zero.

Table 12-1 Command-Line Options for the `authrate` Tool *(Continued)*

Option	Parameter	Purpose
`-R`	*number*	Use randomly generated names in new entries. With this option, `infadd` will first generate the given *number* of random given names and surnames and then randomly select one of each when adding entries. A random name is a sequence of 7 to 12 random letters. When this option is omitted, the tool will use the contents of the `dbgen-GivenNames` and `dbgen-FamilyNames` files in the data subdirectory of the iPlanet DSRK installation.
`-z`	*maxSize*	Specify that all new entries contain an `audio` attribute with a random binary value. The attribute's value is a set of randomly generated bytes, and the number of bytes is randomly chosen in the range [0, *maxSize*].

Command-Line Examples

The examples in this section will measure entry addition performance in your directory server in various scenarios. These examples include sample output. Results will be meaningful only if factors such as machine load and network traffic remain constant during and between tests.

- The *hostname* and *port* placeholders should be replaced with the hostname and port number of your directory.
- For realistic results, thread numbers should be scaled according to your directory's expected load.

Multithreaded Verbose Output

```
$ infadd -h hostname -p port -u "bindDN" -w password \
         -s "ou=people,dc=siroe,dc=com" -t 3 -i 3000 -l 20 -v

Loading Given-Names ...
Loading Family-Names ...
infadd: 3 threads launched.

T1 min:  610ms, max: 1592ms, count:   3, total: 3
T2 min:  702ms, max: 1770ms, count:   2, total: 2
T3 min:  649ms, max: 1308ms, count:   3, total: 3
Average rate:   2.67, total: 8

T1 min:  513ms, max:  607ms, count:   4, total: 7
T2 min:  510ms, max:  655ms, count:   5, total: 7
T3 min:  533ms, max:  721ms, count:   5, total: 8
Average rate:   4.67, total: 22

Total added: 22, Avg rate: 7.33/thrd, 3.67/sec = 272.7msec/op
```

This command launches 3 threads (`-t 3`), each of which binds with the given credentials (`-D "`*bindDN*`" -w` *password*) and adds entries under the `"ou=people,dc=siroe,dc=com"` branch. The name attributes in the new entries are randomly generated from the `Given-Names` and `Family-Names` data files (no `-R` option).

The verbose output (`-v`) gives the measured performance of each thread every 3 seconds (`-i 3000`) until a total of at least 20 new entries have been added (`-l 20`). Each thread reports the following information:

- The minimum and maximum operation times during the elapsed interval.
- The number of operations performed during the elapsed interval.
- The total number of operations performed by that thread so far.

Because multiple threads are used, the average operation rate per thread and the total number of operations is also displayed after every interval. Finally, because of the operation limit option, the summary line is displayed at the end.

Random Strings and Binary Values

```
$ infadd -h hostname -p port -u "bindDN" -w password \
         -s "ou=people,dc=siroe,dc=com" \
         -R 100 -z 10240 -i 3000 -l 20 -v
```

```
Generating random names: 100.  Done.
Generating random names: 100.  Done.
infadd: 1 thread launched.

T1 min:  123ms, max:  241ms, count:   8, total: 8
T1 min:  133ms, max: 2000ms, count:  13, total: 21

Total added: 21, Avg rate: 21.00/thrd, 3.50/sec = 285.7msec/op
```

This command launches a single thread (no `-t` option) that binds with the given credentials (`-D "`*bindDN*`" -w` *password*) and adds entries under the `"ou=people,dc=siroe,dc=com"` branch. The name attributes in the new entries are chosen from 100 randomly generated given names and 100 randomly generated surnames (`-R 100`). The new entries also include the audio attribute with binary values up to 10KB long (`-z 10240`).

The verbose output (`-v`) gives detailed measurements for the thread every 3 seconds (`-i 3000`) until a total of at least 20 new entries have been added (`-l 20`). The data values in the output are the same as described in the previous example.

Part 4

LDIF Deployment Tools

Chapter 13, “dbgen.pl”

Chapter 14, “ldifgen”

Chapter 15, “ldifxform”

Chapter 16, “mmldif”

Chapter 17, “ldiffer.pl”

Chapter 13

dbgen.pl

The dbgen.pl (database generator) tool is a Perl script that generates a sample database containing entries with random values. The output is LDIF (LDAP Data Interchange Format) text and can be loaded into a directory server and used to run performance tests.

As in the ldifgen tool (see Chapter 14), generated entries follow a fixed format, but dbgen.pl uses the standard schema and provides more realistic output. For these reasons, dbgen.pl is the recommended tool for random database creation.

This chapter contains the following sections:

- Command Usage
- Examples and Sample Output

Command Usage

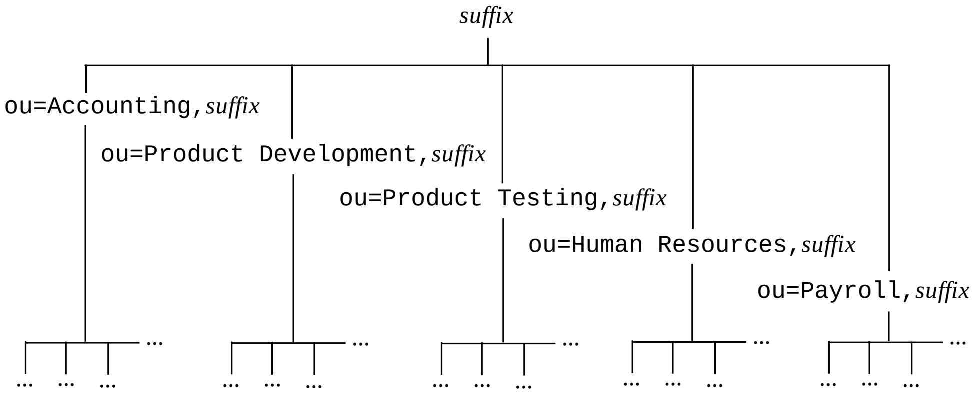
The dbgen.pl tool generates a directory hierarchy that mimics a simple corporate structure. The script has an option for specifying a *suffix*, which will be the root of the generated hierarchy. By default it is dc=siroe,dc=com.

All of the leaf entries are inetOrgPerson objects by default, or you may specify the -O (the capital letter O) option to create all OrganizationalPerson objects. These object classes are used as templates, and the tool generates random values for their allowed attributes. See the “Examples and Sample Output,” on page 138 for examples of both object classes and the generated attribute values.

The tool also generates the root and organizational unit entries so that its output is a complete and valid LDAP directory hierarchy expressed in LDIF text. The organizational unit (ou) entries do not contain any attributes and are always the same. The root entry contains only simple aci attributes (Access Control Instructions), and these are also invariant.

The following diagram shows the directory tree representing the LDIF entries generated by the `dbgen.pl` tool.

Figure 13-1 LDAP Directory Tree Generated by `dbgen.pl`


suffix
ou=Accounting,*suffix*
ou=Product Development,*suffix*
ou=Product Testing,*suffix*
ou=Human Resources,*suffix*
ou=Payroll,*suffix*
...

The script relies on the following files in the *installDir*/`data` directory. You may edit the contents of these files to modify the output of the `dbgen.pl` tool (you will need root privileges to edit these files on UNIX systems):

- **`dbgen-OrgUnits`** - Contains the name of five organizational units. One entry for each organizational unit will be created under the suffix root of the generated directory. Then, each person entry will be randomly placed in one of the organizational units.
- **`dbgen-GivenNames`** and **`dbgen-FamilyNames`** - Contain over 8600 plausible first names and 13400 plausible surnames, respectively. Each entry representing a person is based on a given name and family name, each randomly chosen from these files. A sequential integer is also appended to all names to ensure that no two entries and no two DNs are identical.

Some attribute values such as email addresses are derived from the name and the suffix to create a plausible entry. Other data such as telephone numbers or titles are generated randomly or selected randomly from internal lists. These types of attribute values are not configurable.

NOTE Due to the implementation of randomizing functions on certain platforms, the `dbgen.pl` tool may generate exactly the same attribute values every time it is invoked. While the values are seemingly random, they may in fact be identical to the other entries with the same DN created by different invocations of the command.

Using the `-r` *seed* option will control this phenomenon. Using a different seed with every invocation, such as a timestamp, will ensure that output is always different. Note that invocations that use the same seed may produce the same output.

Syntax

The `dbgen.pl` tool has the following syntax:

```
dbgen.pl -o filename.ldif -n number [options]
```

Where:

- *filename*`.ldif` is a writable file that will contain the LDIF output.
- *number* is the number of leaf entries that will be generated, in addition to the parent entries that are always present.

Running the `dbgen.pl` script without any options or parameters will display the usage help text that briefly describes all options.

The `dbgen.pl` script requires Perl version 5.005_03 or later. See "Where to Find Additional Information," on page 15 for links to Perl resources.

If you customize the `dbgen.pl` script for added functionality, we encourage you to share your work with other LDAP users. Please post a message to the `iplanet.server.idsrk` public newsgroup with your ideas or your code.

Options

The `dbgen.pl` options and parameters are described in the following table.

Table 13-1 Command-Line Options for the `dbgen.pl` Tool

Option	Parameter	Purpose
`-o`	*outputFile*	Specify the output file for the generated database. The output file is a complete directory hierarchy in LDIF text. This parameter is required.
`-n`	*number*	Specify the number of leaf entries in the generated database. This parameter is required.
`-s`	*suffix*	Specify the root of the generated hierarchy. The tool will create an entry of class `top` with a DN equal to the *suffix*, and the DN of all other generated entries will contain this *suffix*. The suffix must begin with either `dc=`*DomainComponent* or `o=`*Organization*. When this option is omitted, the default suffix is `dc=siroe,dc=com`.
`-c`		Use the `cn` (common name) attribute and value in the RDN. When this option is ommited, the `uid` attribte will be used.
`-O`	(capital letter O)	Generate all leaf entries as `OrganizationalPerson` objects and create only the corresponding attributes. When this option is omitted, all leaf entries will have the object class `inetOrgPerson` and the additional attributes of this class.
`-r`	*seed*	Specify a *seed* integer for the random number generator. The data generated by the tool will be different from one execution to the next only if the seed is different. A common way to ensure the seed is different with every execution is to use a timestamp.
`-p`		This option is mentioned in the online help, but it is deprecated.
`-q`		Quiet output mode: `dbgen.pl` will not display any measure of progress while running. When this flag is omitted, the tool will display a line of dots, one dot for every 10,000 entries generated.
`-v`		Verbose output mode: `dbgen.pl` will display additional messages about its progress.

Examples and Sample Output

The examples in this section show the output of the `dbgen.pl` tool. To save space, only the first full leaf entry is shown in each case. These examples use the default data files in the *installDir*`/data` directory.

InetOrgPerson Entries

This example demonstrates how the `dbgen.pl` script generates random attribute values for entries of the `inetOrgPerson` object class. It uses the `date` command of the UNIX shell to generate a seed for the random number generator. It also demonstrates the verbose output that includes progress messages.

```
$ perl dbgen.pl -o out1.ldif -n 3 -r ‘date +%S‘ -v
```

```
Loading Name Data...
Done
Ok, now generating 3 entries, please wait
.Generated 3 entries, 0 duplicates skipped
```

```
$ cat out1.ldif
```

```
dn: dc=siroe,dc=com
objectClass: top
objectClass: domain
dc: siroe
aci: (target=ldap:///dc=siroe,dc=com)(targetattr=*)
 (version 3.0; acl "acl1"; allow(write) userdn = "ldap:///self";)
aci: (target=ldap:///dc=siroe,dc=com)(targetattr=*)
 (version 3.0; acl "acl2"; allow(write) groupdn =
 "ldap:///cn=Directory Administrators, dc=siroe,dc=com";)
aci: (target=ldap:///dc=siroe,dc=com)(targetattr=*)
 (version 3.0; acl "acl3"; allow(read, search, compare) userdn =
 "ldap:///anyone";)

dn: ou=Accounting, dc=siroe,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Accounting

dn: ou=Product Development, dc=siroe,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Product Development

dn: ou=Product Testing, dc=siroe,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Product Testing

dn: ou=Human Resources, dc=siroe,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Human Resources
```

```
dn: ou=Payroll, dc=siroe,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Payroll

dn: uid=HDiogo0, ou=Product Testing, dc=siroe,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Harry Diogo
sn: Diogo
uid: HDiogo0
givenName: Harry
description: This is Harry Diogo0's description
userPassword: HDiogo0
departmentNumber: 5083
employeeType: Manager
homePhone: +1 804 339-8183
initials: H. D.
telephoneNumber: +1 818 605-3216
facsimileTelephoneNumber: +1 206 768-6857
mobile: +1 408 194-1108
pager: +1 415 692-3053
manager: Pippy Mejdal0
secretary: Eden Yvon0
roomNumber: 2265
carLicense: ORHYA5I
l: Redwood Shores
ou: Product Testing
mail: Harry_Diogo@siroe.com
postalAddress: Product Testing Dept #441, Room#73
title: Senior Product Testing Guru

dn: uid=MPlanta1, ou=Product Development, dc=siroe,dc=com
[...]

dn: uid=GVela2, ou=Human Resources, dc=siroe,dc=com
[...]
```

OrganizationalPerson Entries

In this example we use the `-O` (capital letter O) option to generate entries of the `organizationalPerson` object class. We also use the `-c` and `-s` options to customize the DN of the generated entries.

```
$ perl dbgen.pl -o out2.ldif -n 3 -O -r ‘date +%S‘ -q \
                -c -s "o=Varrius Corp.,c=US"

$ cat out2.ldif
dn: o=Varrius Corp.,c=US
objectClass: top
objectClass: organization
o: Varrius Corp.
aci: (target=ldap:///o=Varrius Corp.,c=US)(targetattr=*)
 (version 3.0; acl "acl1"; allow(write) userdn = "ldap:///self";)
aci: (target=ldap:///o=Varrius Corp.,c=US)(targetattr=*)
 (version 3.0; acl "acl2"; allow(write) groupdn =
 "ldap:///cn=Directory Administrators, o=Varrius Corp.,c=US";)
aci: (target=ldap:///o=Varrius Corp.,c=US)(targetattr=*)
 (version 3.0; acl "acl3"; allow(read, search, compare) userdn =
 "ldap:///anyone";)

dn: ou=Accounting, o=Varrius Corp.,c=US
objectClass: top
objectClass: organizationalUnit
ou: Accounting

dn: ou=Product Development, o=Varrius Corp.,c=US
objectClass: top
objectClass: organizationalUnit
ou: Product Development

dn: ou=Product Testing, o=Varrius Corp.,c=US
objectClass: top
objectClass: organizationalUnit
ou: Product Testing

dn: ou=Human Resources, o=Varrius Corp.,c=US
objectClass: top
objectClass: organizationalUnit
ou: Human Resources

dn: ou=Payroll, o=Varrius Corp.,c=US
objectClass: top
objectClass: organizationalUnit
ou: Payroll

dn: cn=Mihaela Inman, ou=Accounting, o=Varrius Corp.,c=US
objectClass: top
objectClass: person
objectClass: organizationalPerson
cn: Mihaela Inman
sn: Inman
uid: MInman0
givenName: Mihaela
```

```
description: This is Mihaela Inman’s description
userPassword: MInman0
telephoneNumber: +1 714 803-8455
facsimileTelephoneNumber: +1 714 371-9310
l: Santa Clara
ou: Accounting
mail: Mihaela_Inman@siroe.com
postalAddress: Accounting Dept #339, Room#98
title: Chief Accounting Director

dn: cn=Randall Braddy,ou=Human Resources,o=Varrius Corp.,c=US
[...]

dn: cn=Morley Cantwell,ou=Product Development,o=Varrius Corp.,c=US
[...]
```

Chapter 14

ldifgen

The ldifgen (LDIF generator) tool creates LDAP entries with randomly generated content. The output is LDIF (LDAP Data Interchange Format) text that can be loaded into an LDAP server to create a test directory. As with the dbgen.pl tool, this generated directory can be used to run performance tests.

This tool creates a simple directory structure with leaf entries belonging to the xyzmember object class. The schema for this custom object class is given by one of the command-line options. You may then use the ldapmodify tool to import both this schema and the generated entries.

This chapter contains the following sections:

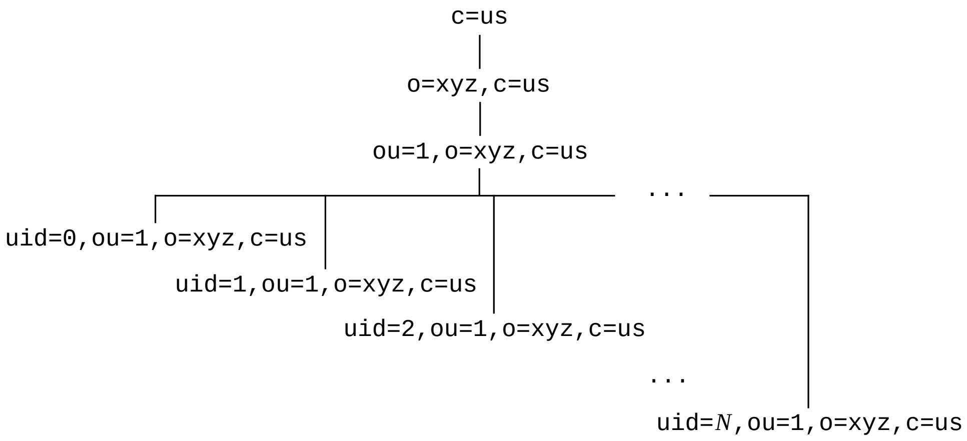
- Command Usage
- Example and Sample Output

Command Usage

The ldifgen tool generates entries that are xyzmember objects. This custom object class acts as a template, and the tool generates random values for each of its attributes. If you do not already have the schema for this object class, use the -S option to generate the version corresponding to your directory. You will need to load this schema entry into the directory before loading the generated entries. An example of this class and its attributes is shown in “Example and Sample Output,” on page 145.

The DNs of generated entries fit within the minimal hierarchy shown in the following diagram. The structure of this tree is not configurable. Only the number of leaf entries is determined by the -n option on the command line.

Figure 14-1 LDAP Directory Hierarchy Created by `ldifgen`



The tool generates the parent entries so that its output is a complete and valid LDAP directory hierarchy expressed in LDIF text. The three parent entries do not contain any attributes and are always the same.

NOTE The DNs of generated entries use sequential numbering to ensure that no two entries of the same invocation will be identical.

However, due to the implementation of randomizing functions on certain platforms, the `ldifgen` tool may generate exactly the same attribute values every time it is invoked. While the values are seemingly random, they may in fact be identical to the other entries with the same DN created by different invocations of the command.

Syntax

The syntax of the `ldifgen` command has the following two forms. The tool uses the standard output, so you need to redirect this to a file in order to save the LDIF text:

```
ldifgen -n number > filename.ldif
ldifgen -S version > filename.ldif
```

Where:

- *number* is the number of leaf entries that will be generated, in addition to the 3 hierarchical entries that are always present.
- *version* is the major version number, either `4` or `5`, of the iPlanet Directory Server instance into which the generated entries will be loaded. Using this option, the `ldifgen` tool will output the corresponding schema for the `xyzmember` class and its attributes. The schema is given as an LDIF update statement that may be loaded using the `ldapmodify` command (see Chapter 4).
- *filename*`.ldif` is a writable file that will contain the LDIF output.

Running the `ldifgen` command without any options or parameters will display the brief usage help text.

Example and Sample Output

The following example shows the output of the `ldifgen` tool. Only the first leaf entry is shown in full, and some attribute values have been shortened to fit on a single line:

```
$ ldifgen -n 3

dn: C=US
c: US
objectclass: top
objectclass: country

dn: O=XYZ,C=US
o: XYZ
objectclass: top
objectclass: organization

dn: OU=1,O=XYZ,C=US
ou: 1
objectclass: top
objectclass: organizationunit

dn: UID=0,OU=1,O=XYZ,C=US
objectclass: top
objectclass: xyzmember
uid: 0
sponsorId: 18080291083208191342756198584726724745154694425751126868
cn: fozepdkdnqlhfejdzksc cfykdxnlorwvfwavbmyq
login: cl1ldo51y5fam665jwlxvhpxdnyuj6z2y6dhl88p
userpassword: 00000000000000000000000000000000000000000000000000000
```

```
postalCode: 0080227
st: jq
cityCode: iwk
city: 49i0zztw1dsrvw3vvcb8siyhghio0aea921ys00z
cityIds: 3yw4lqpm870qv4d1clwi10ig9pk7gaw9zykquf2sc99x3c5l4rtvx22whs
l: tb3y0tuky2
signId: 33382363622400408168917651334147477905498730992720714579805
createTimestamp: 20000101000001Z
modifyTimestamp: 20000101000001Z
mail: cl1ldo51y5fam665jwlxvhpxdnyuj6z2y6dhl88p@xyz.com
topics: gb1cuwjzi0083xogiyleo3kae3w841xaios2r27zsddlg77ovit9rjqvssu
searchFlag: jx2rblz8tddbgg2ndzhy7gep5aqosa0hdtefkjt3mxk93cmn19regb9
myPageTitle: 6wwavgv5qeei1ohbq9ox9l1fn1wohc4tyqtzdeutjy1qcpr8olvnx2
customization: xy01vjqy4qxde49wjrayivb5nalnvg9ykzplplpj1s2ln1nwyn0n
defBookmarks: wa7mz2eudtjkcbb5jornhi0bq5ie5jn7zkj4c3pvny55g6050xi7l
hints: ly8qoxl6r0rk42faell48ahkpvrfy3gp7u6mhxiyngoxotet4p3jpr9ksq5w
birthdate: zbcxbo9jbxn4j27y1bxc76lmnkx6g886qatr48gl
flags: buvl2tptucvr8mkmcnyjlwf1dyjnc5zdpkoh33h4li1kvbcx4gmgirn1vcfx
sourceAppId: 8khnubb4lyewm5m7n4bha3nzlhq6jbahlhbmysw9w0w8vom8j3wzwp
timeZone: 62
c: fo
pageRefresh: 3647
layout: ocra2dp3r1igak6wzkq2bgeo6d8zuog86xpegknebbkrlgtb7938f72rq0g
colors: u356qkqdu4hb5r8heryrrr5w4yu040h59ch62y92sgj3dhqxyeefbaile2c
givenName: fozepdkdnqlhfejdzksc
sn: cfykdxnlorwvfwavbmyq
address: 8jz817ktvj1t5a2ohjc53wrlczwg7evl30zux9tyykhtkqo7f638s0zv5m
phone: 63714139647-45991442
occupation: opsxdliwvjmemuapxydvewsddzwznyfcozztmjsi
householdIncome: 2620083169
gender: F
friend: uid=0,ou=1,o=xyz


dn: UID=1,OU=1,O=XYZ,C=US
objectclass: top
objectclass: xyzmember
uid: 1
...


dn: UID=2,OU=1,O=XYZ,C=US
objectclass: top
objectclass: xyzmember
uid: 2
...
```

Chapter 15

ldifxform

The `ldifxform` (LDIF transform) tool reformats the contents of an LDIF (LDAP Data Interchange Format) text file. It can be used to analyze or edit the contents of a directory offline by processing the LDIF output of the `db2ldif` tool (see Chapter 7, “Command-Line Scripts,” in the *iPlanet Directory Server Command, Configuration and File Reference*).

This tool can perform many transformations on LDIF input, such as converting between all of the most common character sets, extracting attribute values, modifying attribute names, ordering entries based on attribute values, or giving detailed statistics. In all cases, modifications are written to a new LDIF file, and the input file is never modified.

This chapter contains the following sections:

- Command Usage
- Reformatting Commands
- Examples and Sample Output

Command Usage

The `ldifxform` tool acts as a stream filter, reading input from one file, performing any number of transformations and writing the output to another file. Each transformation is specified by a *command* parameter on the command line. Several compatible transformations may be performed simultaneously.

All possible operations are listed in “Reformatting Commands,” on page 149. Some produce LDIF output destined to be reloaded into a directory. For example, renaming an attribute can be more easily processed on an LDIF file than online through requests to a directory server.

Other reformatting operations do not produce LDIF: they are intended to provide an analysis of directory contents. For example, you may extract all different values of a specific attribute and list them under the DN in which they occur. The statistical operations provide counts of entries and attributes.

Syntax

The `ldifxform` tool has the following syntax:

```
ldifxform [-i input.ldif] [-o outputFile] -c "command" ...
```

Where:

- *input*.`ldif` is a readable file that contains the LDIF text input.
- *outputFile* is a writable file that will contain the reformatted LDIF. Some transformations and character conversions will produce output that is not valid LDIF.
- *command* is one of the supported operations to perform on the input, usually enclosed in double quotes ("") for the shell. See “Reformatting Commands,” on page 149 for the list of all available commands. You may specify multiple commands, each preceded by `-c`, if they are mutually compatible.

The input and output parameters are optional, and the tool will use the standard input and output if either or both are omitted. However, the use of files is recommended for portability because standard 8-bit input and output are not fully supported on the Windows platform.

The `ldifxform -h` command will display the usage help text that briefly describes all options.

Options

The `ldifxform` options and parameters are described in the following table.

Table 15-1 Command-Line Options for the `ldifxform` Tool

Option	Parameter	Purpose
`-i`	*input*.ldif	Specify the input file that contains the LDIF text to process. When this option is omitted, the tool will read the standard input.

Table 15-1 Command-Line Options for the `ldifxform` Tool *(Continued)*

Option	Parameter	Purpose
`-o`	*outputFile*	Specify the output file for the reformatted LDIF result. Note that some operations do not produce LDIF output. When this option is omitted, the tool will write to the standard output.
`-c`	*command*	Specify an operation for the tool to apply to the input. The *command* parameter is one of the transformations described in "Reformatting Commands," on page 149. This option may be repeated on the command line when the corresponding operations are compatible.
`-h`		Display the usage help text that briefly describes all options.

Reformatting Commands

The following tables list the type of transformations available through the `-c` *command* parameter of the `ldifxform` tool. The commands are grouped by operations type:

- LDIF Transformations
- Character Set Conversions
- Attribute Name Modifications
- Ordering of Entries
- Directory Statistics

LDIF Transformations

The LDIF transformations affect the encoding and general appearance of LDIF text files.

Table 15-2 LDIF Text Transformations Using `ldifxform`

Command	Formatting Effect
`-c nob64`	Will undo any base-64 transformations. Note that the output will not be reparsable if there are any binary-valued attributes or attributes beginning with special characters.
`-c sevenbit`	Reformats the output as seven-bit characters by base-64 encoding any attribute values that contain non-ASCII bytes. The output is always reparsable.

Table 15-2 LDIF Text Transformations Using `ldifxform` *(Continued)*

Command	Formatting Effect
`-c longlines`	Prevents `ldifxform` from wrapping lines at the 79th column in the output file. This argument is necessary when editing an LDIF file using a UTF-8 aware editor, otherwise a character may be wrapped in the middle of its encoding. The output will be reparsable, but many popular system tools (such as `grep` or `sed`) may not be able to handle lines longer than 1024 characters.
`-c notypes`	Removes attribute type names: the output is no longer LDIF. This is useful for generating a list of values (see "Examples and Sample Output," on page 153).
`-c nodn`	Removes distinguished names: the output is no longer LDIF. This is also useful for generating a list of values.
`-c cleanzero`	Removes trailing zero bytes from attribute values. This option is needed only when processing an LDIF file from a buggy encoder.

Character Set Conversions

The `ldifxform` tool can be used to convert LDIF files to different *charsets* (character sets). Conversions are useful for porting LDIF files between platforms and for use in directories that require localized data. When porting between platforms, you must ensure that all data in the original can be represented in the target charset.

Table 15-3 Character Set Conversions Using `ldifxform`

Command	Formatting Effect
`-c from=88591`	Converts the input file from the ISO-8859-1 charset into the UTF-8 charset. This conversion allows the source data to be written using ISO-8859-1 text editors.
`-c to=88591`	Converts the input file from UTF-8 into the ISO-8859-1 charset. The output is no longer LDIF, and characters that cannot be represented in ISO-8859-1 will be stripped out.
`-c from=t61`	Converts the input file from the T.61 charset into the UTF-8 charset. This option should be used when converting data obtained from an X.500 or LDAPv2 servers that used T.61 charset encoding by default.
`-c to=t61`	Converts the input file from UTF-8 into the T.61 charset. The output is no longer LDIF, and characters that cannot be represented in T.61 will be stripped out.

Table 15-3 Character Set Conversions Using `ldifxform` *(Continued)*

Command	Formatting Effect

`-c from=mstext`
`-c to=mstext`
	This pair of commands converts between UTF-8 and the Windows Unicode Text file format.

`-c from=`*charSet* `-c to=`*charSet*	Additional transformations are supported between UTF-8 and the following platform-specific *charSet*s. Platform-specific transformations will result in data loss for values that cannot be represented in the target charset: **Solaris platform:** 646, 8859-1, 8859-2, 8859-3, 8859-4, 8859-5, 8859-6, 8859-7, 8859-8, 8859-9, 8859-10, eucJP, gb2312, iso2022, KOI8-R, PCK, SJIS, UTF-7, zh_CN.euc, zh_CN.iso2022-7, zh_TW-big5, zh_TW-euc, zh_TW-iso2022-7 **AIX platform:** ASCII-GR, CNS11643.1986-1, CNS11643.1986-2, IBM-1046, IBM-1124, IBM-1129, IBM-850, IBM-856, IBM-932, IBM-eucJP, IBM-eucKR, IBM-eucTW, IBM-sbdTW, IBM-udcJP, IBM-udcTW, ISO8859-1, ISO8859-2, ISO8859-3, ISO8859-4, ISO8859-5, ISO8859-6, ISO8859-7, ISO8859-8, ISO8859-9, JISX0201.1976-0, JISX0208.1983-0, KSC5601.1987-0, TIS-620, big5, ct, fold7, fold8, uucode **HP-UX platform:** roman8, iso8859_1, iso8859_2, iso8859_5, iso8859_6, iso8859_7, iso8859_8, iso8859_9, tis620, eucJP, sjis, big5, ccdc, eucKR, chinese-gb

Attribute Name Modifications

These operations simplify global attribute modifications by replacing or removing attributes for all entries in the LDIF input.

Table 15-4 Attribute Name Modifications Using `ldifxform`

Command	Formatting Effect
`-c suppressoptions`	Remove all options other than binary from attribute type names.
`-c tcut=`*attr*	Remove the attribute named *attr* from all entries where it is found. To remove multiple attributes, specify this command multiple times on the command line.
`-c tpreserve=`*attr*	Remove all attributes except for the given *attr* from all entries. To preserve multiple attributes, specify this command multiple times on the command line.

Table 15-4 Attribute Name Modifications Using `ldifxform` *(Continued)*

Command	Formatting Effect
`-c treplace=`*old*`:`*new*	Replace the *old* attribute type name with the *new* name in all entries where it occurs.

Ordering of Entries

Many directory servers return search results in the order that entries were loaded into the database. By presorting a set of entries known to be static, clients can avoid having to sort results with every query.

Table 15-5 Ordering of LDAP Entries Using `ldifxform`

Command	Formatting Effect
`-c order`	Reorders all entries in the file into hierarchical order.
`-c sort=`*attr*	Sorts entries by increasing value (lowest to highest) of the given *attr* attribute. This is equivalent to alphabetical order for string-valued attributes.
`-c sort=^`*attr*	Sorts entries by decreasing value (highest to lowest) of the given *attr* attribute. This is equivalent to reverse alphabetical order for string-valued attributes.
`-c split=`*N*	Generates multiple LDIF output files that can be loaded into a server by *N* clients in parallel. The output files are named: *outputFile*`_ldifxform_`*c*`_`*n* Where: • *outputFile* is the parameter of the `-o` option and specifies a writable directory and filename prefix. • *c* is the number of components in the root DN of the LDIF file. • *n* is the number of the part, from 1 to *N*.

Directory Statistics

The `ldifxform` tool can be used to analyze the contents of the directory from which the LDIF file is extracted. The output includes a detailed count of DN structures and attribute value occurrences.

Table 15-6 Extracting Directory Statistics Using `ldifxform`

Command	Formatting Effect
`-c stats`	Generates statistical information and appends it as an LDIF comment to the end of the output file.
`-c statsonly`	Generates and outputs only the statistical information. This command is not compatible with any other reformatting or LDIF transformation commands.

Examples and Sample Output

The examples in this section demonstrate the output of the `ldixform` tool. These examples are based the input file `two.ldif` (also used in Chapter 16 in "Examples and Sample Output," on page 159):

```
dn: sn=Jensen,dc=siroe,dc=com
objectclass: top
objectclass: person
cn: Babs Jensen
sn: Jensen
telephoneNumber: 555-5550
createTimestamp: 100

dn: sn=Minsky,dc=siroe,dc=com
objectclass: top
objectclass: person
cn: Pete Minsky
sn: Minsky
telephoneNumber: 555-5559
modifyTimestamp: 200

dn: sn=Morris,dc=siroe,dc=com
objectclass: top
objectclass: person
cn: Ted Morris
sn: Morris
telephoneNumber: 555-5558
createTimestamp: 200
```

The following example shows how to reformat the information in this file so that it is presented as a simple list. The result appears on the standard output because no `-o` option was specified. It gives the ordered list of all telephone numbers assigned to employees of Siroe.com.

Removing the DNs from the output saves space and makes the information more readable. The “sentinel” markers are used internally by the tool and can be edited out of the result if not needed.

```
$ ldifxform -i /export/temp/two.ldif \
            -c "tpreserve=telephonenumber" \
            -c "tpreserve=cn" \
            -c "sort=telephonenumber" \
            -c nodn -c notypes

version: 1
#:ordered: TRUE

objectclass: top
objectclass: sentinel

objectclass: top
objectclass: sentinel

 Babs Jensen
 555-5550

 Ted Morris
 555-5558

 Pete Minsky
 555-5559
```

The following example shows the statistical output of the `ldifxform` tool. The description of the various counters is self-contained within the generated comments.

```
$ ldifxform -i /export/temp/two.ldif -c statsonly

# Basic statistics
#:linecount: 27
#:entrycount: 4
# Number of nonleaf entries (at least one subordinate)
#:nonleafcount: 1
# Number of leaf entries (no subordinates)
#:leafcount: 4
# Largest number of entries immediately below a nonleaf entry
#:maximmsubr: 4
# Number of levels in the DIT hierarchy
#:maxdepth: 3
# Largest number of AVAs in an RDN of an entry’s DN (normally 1)
#:maxrdns: 1
# Attribute types used in the LDIF file
# e is number entries containing this attr, v is total number of
# values, l is total length, m is max length of any one value,
# s is general syntax and x is extra encoding information.
```

```
#:attrstatsinfo: t=deletetimestamp e=1 v=1 l=3 m=3 i=1 s=int
#:attrstatsinfo: t=telephonenumber e=3 v=3 l=24 m=8 i=1 s=tel
#:attrstatsinfo: t=sn e=3 v=3 l=18 m=6 i=1 s=cis x=alphanumeric
#:attrstatsinfo: t=cn e=3 v=3 l=32 m=11 i=1 s=cis x=ascii
#:attrstatsinfo: t=objectclass e=4 v=7 l=38 m=11 i=2 s=cis
#                x=alphanumeric
# Counts of values of specific attribute types
#:attrdomaininfo: t=objectclass v=1 nsTombstone
#:attrdomaininfo: t=objectclass v=3 person
#:attrdomaininfo: t=objectclass v=3 top
# Number of entries with the latest createTimestamp value (UTC)
#:lastaddcount: c=1 t=200
# Number of entries with the latest modifyTimestamp value (UTC)
#:lastmodcount: c=1 t=200
```

Chapter 16

mmldif

The `mmldif` (multi-merge of LDIF) tool combines multiple LDIF files into a single directory hierarchy. The result is the union of all input files: it contains any entry whose DN appears in one or more of the input files.

In a typical usage scenario, `mmldif` can be used to recreate an authoritative database for servers cooperating in a multi-master replication agreement. If for some reason all masters are no longer able to synchronize, the `mmldif` tool can be used to regenerate the master database manually.

This chapter contains the following sections:

- Command Usage
- Examples and Sample Output

Command Usage

The `mmldif` tool performs a union of all input files, based on the DNs they contain: entries that exist in one or more input files are added to the hierarchy of the final output. When the same DN appears in multiple files, the tool will verify that all attribute values are identical. If the attribute values differ, the conflict is resolved by choosing the entry with the most recent time stamp.

The time stamp is given by one of the following attributes: `modifyTimestamp`, `createTimestamp`, or `deleteTimestamp`. The tool also recognizes deleted entries by their `objectclass: nsTombstone` attribute and handles them accordingly. For LDIF files to contain these special attributes, you must use the `db2ldif -r` option when extracting the entries from your directories (see Chapter 7, “Command-Line Scripts,” in the *iPlanet Directory Server Command, Configuration and File Reference*).

Optionally, mmldif can generate change files for each of the input files. Change files contain LDIF update statements that represent the difference between the corresponding input and the merge result. Applying the change file to the directory of the corresponding input will make the directory contents equivalent to the contents of the merge result. To do this, the change file can be used as input to the ldapmodify command (see Chapter 4).

Syntax

The mmldif command has the following syntax:

```
mmldif [-c] [-o output.ldif] input1.ldif input2.ldif ...
```

Where:

- *output*.ldif is the name of a writable file for the LDIF output.
- *inputn*.ldif is an LDIF input file to be merged. You must specify at least two input files, and you may specify more.

The mmldif -h command will display a usage help text that briefly describes all options.

Options

The mmldif options and parameters are described in the following table.

Table 16-1 Command-Line Options for the mmldif Tool

Option	Parameter	Purpose
-o	*outputFile*	Specify the output file for the merge of all LDIF inputs. The output file is itself a complete directory hierarchy in LDIF text. When this parameter is omitted, the mmldif tool produces no output and reports only the number of differences between the input files.
-c		Write a change file containing LDIF update statements for each input file. The change file will have the same path and filename as the corresponding input file, with the addition of the suffix .delta. When using this option, the input files must be in a writable location. Old change files for the same input will be overwritten.
-h		Display the usage help text that briefly describes all options.

Examples and Sample Output

The examples in this section demonstrate the output of the `mmldif` tool. These examples are based the following input files. Differences between the files are shown in bold type:

Table 16-2 Sample `mmldif` Input Files

Filename: `one.ldif`	Filename: `two.ldif`
dn: sn=Jensen,dc=siroe,dc=com	dn: sn=Jensen,dc=siroe,dc=com
objectclass: top	objectclass: top
objectclass: person	objectclass: person
cn: Babs Jensen	cn: Babs Jensen
sn: Jensen	sn: Jensen
telephoneNumber: 555-5550	telephoneNumber: 555-5550
createTimestamp: 100	createTimestamp: 100
dn: sn=Minsky,dc=siroe,dc=com	dn: sn=Minsky,dc=siroe,dc=com
objectclass: top	objectclass: top
objectclass: person	objectclass: person
cn: Pete Minsky	cn: Pete Minsky
sn: Minsky	sn: Minsky
telephoneNumber: **555-5551**	telephoneNumber: **555-5559**
createTimestamp: **100**	modifyTimestamp: **200**
dn: sn=Rose,dc=siroe,dc=com	**dn: sn=Morris,dc=siroe,dc=com**
objectclass: top	**objectclass: top**
objectclass: person	**objectclass: person**
cn: Paula Rose	**cn: Ted Morris**
sn: Rose	**sn: Morris**
telephoneNumber: 555-5552	**telephoneNumber: 555-5558**
createTimestamp: 100	**createTimestamp: 200**
	dn: sn=Rose,dc=siroe,dc=com
	objectclass: nsTombstone
	deleteTimestamp: 200

Merge Statistics

The `mmldif` tool always displays merge statistics to the standard output. The `-o` *outputFile* parameter may be omitted when you are interested only in the number of differences between input files.

```
$ mmldif one.ldif two.ldif

start time Wed Jul 11 12:34:56 2001

entry counts: unchanged=1 changed=2 new=1 total=4

end time Wed Jul 11 12:34:56 2001
differencing took <= 1 second
```

This output confirms what we can see in our simple example files:

- `unchanged=1` - There is one entry where all attributes match exactly.
- `changed=2` - Two entries with matching DNs have different attribute values.
- `new=1` - One entry is considered new: its DN does not appear in all files.
- `total=4` - The input files contain a total of four different DNs.

Merge Output File

Running the command again with the `-o` *outputFile* parameter will allow us to see the result of the merge.

```
$ mmldif -o merge.ldif one.ldif two.ldif

start time Wed Jul 11 12:34:56 2001

entry counts: unchanged=1 changed=2 new=1 total=4

end time Wed Jul 11 12:34:56 2001
differencing took <= 1 second

$ cat merge.ldif

dn: sn=Jensen,dc=siroe,dc=com
cn: Babs Jensen
CreateTimestamp: 100
objectclass: person
objectclass: top
sn: Jensen
telephoneNumber: 555-5550

dn: sn=Minsky,dc=siroe,dc=com
cn: Pete Minsky
objectclass: person
objectclass: top
sn: Minsky
telephoneNumber: 555-5559
```

```
dn: sn=Morris,dc=siroe,dc=com
cn: Ted Morris
CreateTimestamp: 200
objectclass: person
objectclass: top
sn: Morris
telephoneNumber: 555-5558
```

This output shows the result of the union of the input files:

- The first entry, `sn=Jensen`, remains unchanged.
- Only the telephone number is modified in the second entry, `sn=Minsky`.
- The third entry, `sn=Morris`, is counted as new because it was not present in `one.ldif`.
- The deleted entry, `sn=Rose`, is not included, although it is tallied as a change in the entry counts.

Change Files

In order to see how each input relates to the merge result, we can generate the change files. These will contain LDIF update statements showing the difference between each input file and merge result shown in the previous example:

```
$ mmldif -o merge.ldif -c one.ldif two.ldif

start time Wed Jul 11 12:34:56 2001
entry counts: unchanged=1 changed=2 new=1 total=4
end time Wed Jul 11 12:34:56 2001
differencing took <= 1 second

$ cat one.ldif.delta

dn: sn=Minsky,dc=siroe,dc=com
changetype: modify
delete: createTimestamp
-
replace: telephoneNumber
telephoneNumber: 555-5559
-

dn: sn=Rose,dc=siroe,dc=com
changetype: delete
```

```
dn: sn=Morris,dc=siroe,dc=com
changetype: add
cn: Ted Morris
createTimestamp: 200
objectclass: person
objectclass: top
sn: Morris
telephoneNumber: 555-5558
```

The file `two.ldif.delta` is also generated but it is empty (zero length) because `two.ldif` contains all of the most recent modifications between the two input files.

This output shows how to bring the contents of `one.ldif` up to date with the result of the merge:

- Modify the entry for `sn=Minsky` by:
    - Deleting the `createTimestamp` attribute: `mmldif` removes timestamp attributes from modified entries.
    - Replacing the value of the `telephoneNumber` attribute.
- Delete the entire entry for `sn=Rose`.
- Add the entire new entry for `sn=Morris`.

Supposing that `one.ldif` represents the contents the directory on host `one`, port `389`, the following command will update those contents so that they are equivalent to `merge.ldif`:

```
$ ldapmodify -h one -D "bindDN" -w bindPassword \
             -f one.ldif.delta
```

This command uses the credentials of the *bindDN* to access the directory and assumes that user has permission to modify entries. See Chapter 4, “ldapmodify,” for more information.

Chapter 17

ldiffer.pl

The `ldiffer.pl` (LDAP difference) tool is a Perl script that detects differences between two LDAP directories and can perform modifications accordingly. Comparisons are based on attribute values so that directories with different DN structures and different schemas may be compared.

When the tool detects comparable entries in each directory, it can be configured to add or modify attributes in either entry. The value of new or modified attributes is given as a regular expression involving other attribute values. This flexibility can be used to migrate data to a new schema or to import specific data from one directory into another.

This chapter contains the following sections:

- Command Usage
- Configuration File Format
- Configuration File Example

Command Usage

The `ldiffer.pl` tool interacts with two directories. One is called "sequential" and can be considered the source of information. The other is called "random" and can be considered the target to be modified. However, the tool also allows some modifications to be made in the source directory.

The script uses the `ldapsearch` tool (see Chapter 3) to extract a set of entries from both directories. The bind credentials, the base DN, and the filter for each search are specified separately, meaning that arbitrarily different sets of entries may be compared.

Entries from each set are matched based on equality between the value of one or more key attributes. This allows you to match entries that do not necessarily have the same DN. Because only values are compared, key attributes may also have different names in each directory.

Within a pair of matched entries, the tool then checks the values of one or more comparison attributes. A discrepancy between any of the comparison attribute values will trigger the tool to perform a set of update actions. An update action involves modifying an attribute in one of the entries based on the value of an attribute in its matched entry, after processing that value with a regular expression. The ldiffer.pl script uses the `ldapmodify` tool (see Chapter 4) to modify entries.

For example, an administrator may wish to migrate information from an old directory under the suffix `o=Siroe.com,c=us` to a new directory under `dc=Siroe,dc=com`. The key attribute might be employee login names that would be invariant. Then, `ldiffer.pl` could compare an attribute such as `fax` in a custom schema to `facsimileTelephoneNumber` in the standard schema of the new directory. Finally, the value of `facsimileTelephoneNumber` can be updated with the value of the `fax` attribute and with, for example, a new area code.

When matching entries are not found, the tool may also be configured to either delete the entry in the source directory, add it to the target directory, or both.

Due to the flexibility it offers, the `ldiffer.pl` script requires many parameters to describe the modifications of one or both target directories. These parameters are given in a configuration file that is named on the command line.

Syntax

The `ldiffer.pl` tool has the following syntax:

`ldiffer.pl` *configFile*

Where:

- *configFile* is the name of a file that contains all of the configuration parameters for binding to the directories, performing comparisons, and how to make any potential modifications. These parameters are described in “Configuration File Format,” on page 165.

All output from the script is written to log files with names and locations based on that of the configuration file:

- *configFile*.*timestamp*`.action.log` - Displays the comparisons, additions, and modifications that were performed in a given run. Also, gives a count of the differences and lists the entries that were not successfully matched.

- *configFile*.*timestamp*.debug.log - Records the commands that were executed while comparing and modifying the directories. This file is created only if the debugging parameter is specified.

The ldiffer.pl script requires Perl version 5.005_03 or later. See "Where to Find Additional Information," on page 15 for links to Perl resources.

If you customize the ldiffer.pl script for added functionality, we encourage you to share your work with other LDAP users. Please post a message to the iplanet.server.idsrk public news group with your ideas or your code.

Configuration File Format

The configuration file uses the following syntax:

```
# line of comment
parameter:value
...
```

The parameters may be given in any order, but are usually grouped according to the directory they affect. See the sample configuration file in "Configuration File Example," on page 169.

You will need to specify values for all of the parameters in the configuration file before running ldiffer.pl. Due to the potential complexity of comparisons and updates, we recommend testing your configuration files on mock directories with the debugging parameter turned on.

The global parameters in the following table affect the behavior of the tool.

Table 17-1 Global Parameters in the ldiffer.pl Configuration File

Parameter	**Purpose**
ldapsearch	Specify the path for running the ldapsearch tool. If you installed the iPlanet DSRK in the default location, this parameter should be set to /opt/iPlanet/bin/idsrk/ldapsearch.
ldapmodify	Specify the path for running the ldapmodify tool. If you installed the iPlanet DSRK in the default location, this parameter should be set to /opt/iPlanet/bin/idsrk/ldapmodify.
debug	Indicate the use of the debugging log. The possible values are 0 for no logging or 1 for creating the log file.

Table 17-1 Global Parameters in the `ldiffer.pl` Configuration File *(Continued)*

Parameter	Purpose
`delimiter`	Set the character used internally to delimit key attribute values. You should set this parameter to a character that does not occur in the attribute values of your directories, for example `!`.

All other parameters have one of the following prefixes:

- **`sqn_`** for parameters that apply to the "sequential," or source, directory.
- **`rnd_`** for parameters that apply to the "random," or target, directory.

The connection parameters determine the LDAP directories to operate upon and give the bind credentials for accessing them.

Table 17-2 Connection Parameters in the `ldiffer.pl` Configuration File

Parameter	Purpose

`sqn_h`
`sqn_p`
`rnd_h`
`rnd_p`
	Specify the host name or IP address and port number of each LDAP directory.

`sqn_D`
`sqn_w`
`rnd_D`
`rnd_w`
	Give the bind DN and password needed to access each directory. These parameters may be left blank for anonymous binding. The bind credentials must have permissions for both searching the directories and modifying entries, if necessary. Also note that the bind DN may influence the results of searches when determining the set of entries to compare.

The comparison parameters determine the set of entries to compare, how to match entries from each directory, and the attribute values that will trigger actions.

Table 17-3 Comparison Parameters in the `ldiffer.pl` Configuration File

Parameter	Purpose

`sqn_b`
`sqn_filter`
`rnd_b`
`rnd_filter`
	Specify the base DN and filter string used to search each directory. The filter string should be an RFC 2254-compliant LDAP search filter (see "LDAP Search Filters" in Appendix B of the *iPlanet Directory Server Administrator's Guide*), for example: `uid=*`
	These parameters determine the set of entries that will be compared from each directory. These parameter values are not necessarily identical, as the `ldiffer.pl` tool allows you to match entries based on any attribute values, not only the DN of an entry.

`sqn_key_attrs` `rnd_key_attrs`	Specify the names of the key attributes whose values must be equal in order to match an entry from each directory. These parameters may contain multiple attribute names in a comma separated list. These lists may contain different attribute names, but they must contain the same number of names. In order for two entries to match, all attribute values must match in the order that the attribute names are given in their respective list. To match entries based on their DN, specify the `dn` attribute in both parameters.
`sqn_comp_attrs` `rnd_comp_attrs`	Specify the names of the comparison attributes for entries from each directory. These are also comma separated lists containing an equal number of attribute names. When two entries match according to the key attributes, the values of the comparison attributes are compared, again in the order that the attribute names are given in their respective list. If any one of the comparison attribute values differ in the pair, the two entries will be modified according to the action parameters. To perform a DN-based comparison, specify the `dn` attribute in both lists. If any one of the attributes does not exist in its respective entry, the comparison will fail but no action will be triggered. If you wish to trigger an action for every pair of matching entries, you must specify a single attribute name in each list parameter that you know will have a value and be different on each entry.

The action parameters in the following table determine what modifications will be made when matching entries fail the comparison or when entries fail to match. These parameters are complex lists of the following format:

Statement1`,`*Statement2*`,...`

Where each *Statement* has the following syntax:

AttributeName;*otherAttribute*;*regularExpression*

The *regularExpression* must follow the syntax for Perl regular expressions. Regular expressions may also be omitted, but the punctuation of the list must be respected, for example:

```
givenName;firstName;,sn;lastName;
```

Table 17-4 Action Parameters in the `ldiffer.pl` Configuration File

Parameter	Purpose
`sqn_add_attrs` `rnd_add_attrs`	Specify the attributes to add to each of the matched entries and how to determine their initial value. These parameters are complex lists of the format described previously. Each *Statement* will create a new attribute with the given *AttributeName* in the entry of the corresponding directory (either `sqn` or `rnd`). The initial value given to the attribute will be the value of the *otherAttribute* in the matching entry (in the other directory), after applying the *regularExpression*. To create multivalued attributes, specify multiple add statements with the same *AttributeName*.
`sqn_mod_attrs` `rnd_mod_attrs`	Specify the attribute values to modify and how to determine their new value. These parameters have the same complex list format as described previously. Each *Statement* will modify the value of the given *AttributeName* in the entry of the corresponding directory (either `sqn` or `rnd`). The new value will be that of the *otherAttribute* in the matching entry (in the other directory), after applying the *regularExpression*.

All actions specified by parameters in the previous table will be triggered when a pair of matching entries fail the comparison. Thus, you may configure `ldiffer.pl` to add attributes or modify entries in both directories. Leave parameters blank if that specific action is not required. For example, if you do not wish to modify the source “sqn” directory, do not define the `sqn_add_attrs` or `sqn_mod_attrs` parameters.

The tool currently offers less flexibility for entries that do not match. After pairing up all entries in the chosen set from each directory, `ldiffer.pl` can only determine those entries from the source “sqn” directory that do have a match. These entries may optionally be removed from the “sqn” directory, added to the “rnd” directory, or both, according to the parameters in the following table.

Table 17-5 Unmatched Entry Parameters in the `ldiffer.pl` Configuration File

Parameter	Purpose
`rnd_add_unmatched`	Specify whether unmatched entries from the source “sqn” directory will be added to the target “rnd” directory. The value of this flag is either `0` for no additions or `1` for adding all unmatched entries. When entries are added, there is no mechanism to modify their attributes or attribute values. Therefore, when this flag is true (`1`), the target “rnd” directory must support the same DN hierarchy and the same schema as the source “sqn” directory.
`sqn_del_unmatched`	Specify whether unmatched entries are deleted from the source “sqn” directory. The value of this flag is either `0` to leave the “sqn” directory unchanged or `1` to remove all unmatched entries.

Configuration File Example

The iPlanet DSRK installation includes a configuration file example named *installDir*`/perl/ldiffer-sample.config`:

```
# Global parameters
ldapsearch:/opt/iPlanet/bin/idsrk50/ldapsearch
ldapmodify:/opt/iPlanet/bin/idsrk50/ldapmodify
debug:1

# the delimiter should be a character that will not
# occur in the values of key attributes to match
delimiter:!

# The "sequential" directory can be considered the source
sqn_h:legacy.siroe.com
sqn_p:389
sqn_D:cn=directory manager
sqn_w:bindPassword
sqn_b:o=Siroe.com,c=us
sqn_filter:login=*
sqn_key_attrs:login
sqn_comp_attrs:dn
sqn_add_attrs:branch;l;
sqn_mod_attrs:
sqn_del_unmatched:0
```

```
# The "random" directory can be considered the target
rnd_h:ldap.siroe.com
rnd_p:389
rnd_D:cn=directory manager
rnd_w:bindPassword
rnd_b:dc=siroe,dc=com
rnd_filter:uid=*
rnd_key_attrs:uid
rnd_comp_attrs:dn
rnd_add_attrs:
rnd_mod_attrs:mail;email;,cn;fullName;,sn;fullName;s/.*\s//
rnd_add_unmatched:0
```

This configuration example might be used in a hypothetical scenario where Siroe.com has undergone geographic expansion and has created a new LDAP directory using the standard schema. Some data has already been added to the new directory, such as employee `uid` and location (`l`). The rest of the needed information can be extracted from the legacy directory that has a custom schema.

Running the `ldiffer.pl` tool with this configuration file will match entries based on the `uid` attribute. Then, data from the legacy directory can be used to set attributes with standard names (`mail`, `cn`, and `sn`) in the new directory. If these attributes do not exist in the new directory, they will be created. The value of the surname (`sn`) attribute is assumed to be the second word of the legacy `fullName` attribute, as extracted by the Perl regular expression `s/.*\s//`.

However, the legacy system is still in use by the human resources department, and now that Siroe.com has several offices, it needs to update the legacy directory. The tool will copy the employees' new locations into a new attribute called `branch`.

After the migration, Siroe.com might also wish to keep information in the legacy directory up to date, assuming that the latest modifications occur only in the new directory. The `ldiffer.pl` tool can be used to automate the transfer of information back into the schema of the legacy directory. Another configuration file would be needed for this scenario, in which the attribute values in the new "rnd" directory would be used to modify attributes in the legacy "sqn" directory.

Updating or creating new directories can be highly automated through the use of the `ldifxform` (see Chapter 15) and `ldiffer.pl` tools. Together, they can modify extracted information for complex operations such as porting existing data to a new schema. However, careful planning is necessary to perform a complicated sequence of transformations. Directory modifications of this scope will require special considerations such as turning off schema checking and disallowing other read and write requests while the data is in an intermediate state.

Part 5

Maintenance and Debugging Tools

Chapter 18

logconv.pl

The logconv.pl (log converter) tool is a Perl script that analyzes the access logs of an iPlanet Directory Server to extract usage statistics and count the occurrences of significant events. It is compatible with log formats from iPlanet Directory Server 3.*x*, 4.*x*, and 5.*x*.

The logconv.pl script requires Perl version 5.005_03 or later. See “Where to Find Additional Information,” on page 15 for links to Perl resources.

If you customize the logconv.pl script for added functionality, we encourage you to share your work with other LDAP users. Please post a message to the iplanet.server.idsrk public newsgroup with your ideas or your code.

This chapter contains the following sections:

- Command Usage
- Sample Output

Command Usage

The tool will extract the following information from access logs:

- Number of restarts
- Total number of connections
  Total operations requested
  Total results returned
  Results to requests ratio
- Number of searches
  Number of modifications
  Number of adds
  Number of deletes
  Number of modified RDNs
- For iPlanet DS 5.*x* logs only:
  Persistent searches
  Internal operations (with verbose logs)
  Entry operations (with verbose logs)
  Extended operations
  Abandoned requests
  Smart referrals received (verbose logs)
- VLV (virtual list view) operations
  VLV unindexed searches
  Server-side sorting operations
  SSL connections
- Performance lowering operations:
  Entire database searches
  Unindexed searches (details optional)
- FDs (file descriptors) taken
  FDs returned
  Highest FD taken
- Disruptions:
  Broken pipes
  Connections reset by peer
  Unavailable resources (and detail)
- Total binds and types of binds
- Most frequent occurrence lists (optional):
  Error and return codes
  Failed logins
  Connection codes
  Client IP addresses and connection codes
  Bind DNs
  Base DNs for searching
  Search filters
  Etimes (elapsed operation time)
  Longest etimes
  Nentries (number of entries in result)
  Largest Nentries
  Extended operations (DS 5.*x* only)
  Most requested attributes (DS 5.*x* only)
  Abandoned operation details (DS 4.15)
- Recommendations (optional)

The `logconv.pl` tool displays two types of statistics that administrators will find useful for monitoring and optimizing directory usage:

- Simple counts of events such as the total number of binds and the total number of searches provide overall usage information. This is the basic information that the tool will always print (see “Sample Output,” on page 177).
- Lists of the most frequently occurring parameters in LDAP requests provide insight how the directory information is being accessed. For example, lists of the top ten bind DNs, base DNs, filter strings, and attributes returned can help administrators optimize the directory for its users. These lists are optional because they are computation intensive: specify only the command-line options for those you need (see “Options,” on page 176).

Some information that is extracted by the `logconv.pl` script is available only in iPlanet Directory Server 5.0 logs: the corresponding values will be zero when analyzing logs from other versions. In addition, some information will only be present in the logs if verbose logging is enabled in your directory server. For more information, see “nsslapd-accesslog-level” in the *iPlanet Directory Server Command, Configuration and File Reference*.

The following issues will affect the output and performance of this tool:

- Some data extracted from logs depend on connection and operation numbers that are reset and no longer unique after a server restarts. Therefore, to obtain the most accurate counts, the logs to be analyzed should not span the restart of the directory server.
- Due to changes in access logs formats in Directory Server 5.0 that also affect operation numbers, the tool will be more accurate on 5.0 logs when processing large amounts of access logs.
- For performance reasons, it is not recommended to run more than one gigabyte of access logs through the script at any one time.

The `logconv.pl` script is supported when using Perl version 5.005_03. See “Where to Find Additional Information,” on page 15 for links to Perl resources.

If you customize the `ldiffer.pl` script for added functionality, we encourage you to share your work with other LDAP users. Please post a message to the `iplanet.server.idsrk` public newsgroup with your ideas or your code.

Syntax

The `logconv.pl` tool has the following syntax:

```
logconv.pl [options] [-efcibaltnxgju] accessLog ...
```

Where:

- *options* and `[-efcibaltnxgju]` are the command-line options described in the next section.
- *accessLog* is the name of a file that contains the access log of your iPlanet Directory Server. You may use wildcards in the filename or specify multiple filenames. However, the statistics are computed over the set of all logs, so all logs should pertain to the same directory server. The tool will ignore any file with the name `access.rotationinfo`.

The `logconv.pl -h` command will display the usage help text that briefly describes all options.

Options

The `logconv.pl` command-line options are described in the following table.

The parameters without a preceding dash (`-`) at the end of the table will enable the optional lists of occurrences. Specify only those you need to limit the output and improve execution speed. You may specify any number of these parameters in any order, but they must all be given together as a single option on the command line, for example: `-abcefg`.

Regardless of the order of options on the command line, the lists will appear in the output in the order they are listed in this table. Use the `-V` option to display all optional output. Also, use the `-s` *number* option to control the length of these lists.

Table 18-1 Command-Line Options for the `logconv.pl` Script

Option	Parameter	Purpose
`-d`	*mgrDN*	Specify the DN (distinguished name) of the directory manger in the logs being analyzed. This allows the tool to collect statistics for this special user. The *mgrDN* parameter should be given in double quotes (`""`) for the shell. When this parameter is omitted, `logconv.pl` will use the default manager DN of iPlanet Directory Server: `"cn=directory manager"`.
`-X`	*IPaddress*	Specify the IP address of a client to exclude from the statistics. This client will not appear in lists of IP addresses (the `i` flag), and the connection codes it generates will not be tallied in the total connections (default statistic) nor in the connection code details (the `c` flag). For example, you may wish to ignore the effect of a load balancer that connects to the directory server a regular intervals. This option may be repeated to exclude multiple IP addresses.
`-v`		Display the version number of the `logconv.pl` script.
`-h`		Display the usage help text that briefly describes all options.
`-s`	*number*	Specify the number of items in each of the list options below. The default is `20` when this parameter is omitted. For example, `-s 10 -i` will list the ten client machines that access the directory server most often. This parameter will apply to all lists that are enabled, and it will have no effect if none are displayed.
`-V`		Enable the most verbose output. With this option, `logconv.pl` will compute and display all of the optional lists described below.
`e`		List the most frequent error and return codes.
`f`		List the bind DNs with the most failed logins (invalid password).
`c`		List the number of occurrences for each type of connection code.

Table 18-1 Command-Line Options for the `logconv.pl` Script *(Continued)*

Option	Parameter	Purpose
`i`		List the IP addresses and connection codes of the clients with the most connections, which detects clients that may be trying to compromise security.
`b`		List the most frequently used bind DNs.
`a`		List the most frequent base DNs when performing operations.
`l`		List the most frequently used filter strings for searches.
`t`		List the longest and most frequent etimes (elapsed operation time)
`n`		List the largest and most frequent nentries (entries per result).
`x`		List the number and OID of all extended operations (DS 5.*x* only).
`r`		List the names of the most requested attributes (DS 5.*x* only).
`g`		List the details of all abandoned operations.
`j`		Give recommendations based on data collected from the log file.
`u`		Give operation details about unindexed searches.

Sample Output

The following example shows the verbose output (`-V`) of the `logconv.pl` tool. It will read all access logs in the logs directory, ignoring `access.rotationinfo` files. After processing the log files, it displays all of the access statistics and event counters. Then it shows all of the lists of most frequent connection and operation values, with the top 10 in each category (`-s 10`). It ends with a set of general recommendations triggered by certain values or events.

```
$ perl logconv.pl -V -s 10 \
                  /usr/iplanet/servers/slapd-serverID/logs/access*

verbose output enabled

Log Analyzer 4.11

Initializing Variables...

Processing 3 Access Log(s)...
```

```
access (Total Lines: 5870)
        1000 Lines Processed
        2000 Lines Processed
        3000 Lines Processed
        4000 Lines Processed
        5000 Lines Processed
*       5870 Lines Processed      Total Lines Processed:  5870

access.20010713-130613 (Total Lines: 7912)
        1000 Lines Processed
        2000 Lines Processed
        3000 Lines Processed
        4000 Lines Processed
        5000 Lines Processed
        6000 Lines Processed
        7000 Lines Processed
*       7912 Lines Processed      Total Lines Processed: 13782

access.20010714-150617 (Total Lines: 6338)
        1000 Lines Processed
        2000 Lines Processed
        3000 Lines Processed
        4000 Lines Processed
        5000 Lines Processed
        6000 Lines Processed
*       6338 Lines Processed      Total Lines Processed: 20120

* Total Lines Analyzed: 20120


----------- Access Log Output ------------

Start of Log:  18/Jul/2001:13:08:18
End of Log:    18/Jul/2001:17:05:07

Restarts:                     1

Total Connections:             4002
Total Operations:             14818
Total Results:                14908
Overall Performance:          100.6%

Searches:                     4354
Modifications:                27
Adds:                         26
Deletes:                      30
Mod RDNs:                     0
```

```
5.x Stats
Persistent Searches:          1
Internal Operations:          0
Entry Operations:             0
Extended Operations:          6935
Abandoned Requests:           29
Smart Referrals Received:     0

VLV Operations:               49
VLV Unindexed Searches:       49
SORT Operations:              44
SSL Connections:              0

Entire Search Base Queries:   3912
Unindexed Searches:           1

  Unindexed Search #1
  -  Date/Time:             18/Jul/2001:13:33:19
  -  Connection Number:     2926
  -  Operation Number:      1
  -  Etime:                 0
  -  Nentries:              4001
  -  IP Address:            192.18.122.229
  -  Bind DN:               cn=directory manager
  -  Search Filter:         (objectclass=*)

FDs Taken:                    3448
FDs Returned:                 3446
Highest FD Taken:             89

Broken Pipes:                 0
Connections Reset By Peer:    0
Resource Unavailable:         1
  -  1    (T1) Idle Timeout Exceeded

Binds:                        3446
Unbinds:                      3438

 LDAP v2 Binds:               1
 LDAP v3 Binds:               3445
 SSL Client Binds:            0
 Failed SSL Client Binds:     0
 SASL Binds:                  1
  1     DIGEST-MD5

 Directory Manager Binds:     16
 Anonymous Binds:             1
 Other Binds:                 3429


----- Errors -----
```

```
err=0                 14737    Successful Operations
err=32                   75    No Such Object
err=12                   62    Unavailable Critical Extension
err=10                    3    Referral Received
err=49                    1    Invalid Credentials (Bad Password)
err=65                    1    Objectclass Violation


----- Top 10 Failed Logins ------

1           uid=rmanager,cn=config


----- Total Connection Codes -----

U1                     3437    Cleanly Closed Connections
B1                        8    Bad Ber Tag Encountered
T1                        1    Idle Timeout Exceeded


----- Top 10 Clients -----

Number of Clients:  2

3440   123.456.789.001
                  3429 -  U1   Cleanly Closed Connections
                     8 -  B1   Bad Ber Tag Encountered
                     1 -  T1   Idle Timeout Exceeded

8      127.0.0.1
                     8 -  U1   Cleanly Closed Connections


----- Top 10 Bind DN’s -----

Number of Unique Bind DN’s: 8

3422            uid=rmanager,cn=config
14              cn=dm
5               uid=aa,cn=config
1               Anonymous Binds
1               uid=rmanager
1               cn=dma,cn=config
1               dc=dm
1               cn=dma


----- Top 10 Search Bases -----

Number of Unique Search Bases: 73
```

```
3519            root dse
256             ou=people,dc=siroe,dc=com
82              cn=ldbm database, cn=plugins, cn=config
57              cn=monitor
51              dc=siroe,dc=com
48              cn=config
30              cn=mapping tree,cn=config
28              cn=Babs Jensen,ou=peopled,c=siroe,dc=com
22              cn=plugins,cn=config
20              cn=features,cn=config


----- Top 10 Search Filters -----

Number of Unique Search Filters: 31

3502            (objectclass=*)
408             (|(objectclass=*)(objectclass=ldapsubentry))
119             (uid=*)
88              (objectclass=nsbackendinstance)
6               (nsslapd-backend=userroot)
6               (nsslapd-plugintype=database)
4               (uid=bjensen)
4               (objectclass=subschema)
4               (objectclass=nsindex)
3               (cn=config)


----- Top 10 Most Frequent etimes -----

14634           etime=0
229             etime=1
9               etime=2
3               etime=7
1               etime=8
1               etime=3
1               etime=4
1               etime=5


----- Top 10 Longest etimes -----

etime=8         1
etime=7         3
etime=5         1
etime=4         1
etime=3         1
etime=2         9
etime=1         229
etime=0         14634
```

```
----- Top 10 Largest nentries -----

nentries=25                     5
nentries=11                     2
nentries=10                     2
nentries=9                      2
nentries=8                      1
nentries=5                      1
nentries=4                     16
nentries=3                     62
nentries=2                     37
nentries=1                   3986
nentries=0                    239


----- Top 10 Most returned nentries -----

3986            nentries=1
239             nentries=0
62              nentries=3
37              nentries=2
16              nentries=4
5               nentries=25
2               nentries=10
2               nentries=11
2               nentries=9
1               nentries=5


----- 5.x Extended Operations -----

3454    2.16.840.1.113730.3.5.3    Start Replication Request
                                     (incremental update)

3438    2.16.840.1.113730.3.5.5    End Replication Request
                                   (incremental update)

43      2.16.840.1.113730.3.5.6    Replication Entry Request


----- Top 10 Most Requested Attributes -----

3420        supportedControl
3420        supportedExtension
360         All Attributes
341         numSubordinates
328         objectClass
315         nsAccountLock
144         nsBackendSuffix
```

```
104          nsslapd-suffix
36           dn
32           cn


----- Abandon Request Stats -----

 - SRCH conn=2 op=10 msgid=1092 client=127.0.0.1
 - BIND conn=2 op=0 msgid=1119 client=127.0.0.1


----- Recommendations -----

 1.  You have unindexed searches, this can be caused from a search on
a unindexed attribute, or your returned results exceeded the
allidsthreshold.  Unindexed searches are not acceptable, please make
any configuration changes necessary to resolve these searches!

 2.  You have some connections that are are being closed by the
idletimeout setting.  You may want to increase the idletimeout if it
is set low.

 3.  You have a high number of searches that query the entire search
base.  Although this is not necessarily bad, it could be resource
intensive if the search base contains many entries.
```

There are many possible recommendations depending on the statistics and occurrences of certain events. The recommendations are based on general administration guidelines and should be adapted to fit the actual usage of your directory server.

Chapter 19

migrateSchemaTo5.pl

The `migrateSchemaTo5.pl` tool is a Perl script that helps automate the process of updating your schema for iPlanet Directory Server 5.*x*.

This script requires Perl version 5.005_03 or later. See “Where to Find Additional Information,” on page 15 for links to Perl resources.

The tool’s usage text in the following output shows the command-line syntax:

```
Usage:
migrateSchemaTo5.pl -o 4.xInstancePath [-i oldSchemaFile]
                    -s newSchemaFile  [-x X-ORIGIN]
                    [-t tracelevel] [-L logfile]

** parameters in brackets are optionals, others are required **

-o 4.xInstancePath - Path of the 4.x instance to migrate. Will
                     read slapd.conf file for userat and
        or           useroc settings
-i oldSchemaFile   - Old schema file to convert instead of
                     files found in userat and useroc variables
                     in slapd.conf. Must be a full path to file
                     This parameter will override the -o option
-s newSchemaFile   - New DS5 Schema file for converted custom
                     4.x schema
[-x X-ORIGIN]      - Specify the X-ORIGIN value setting for
                     attributes and objectclasses. Default is
                     ’user defined’
[-t tracelevel]    - Specify the level of trace (0..4)
[-L logfile]       - Specify the file to log the migration report
```

Further documentation for this tool was not available at the time of publication. Please look for the latest version of this guide at the following URL:

```
http://docs.iplanet.com/docs/manuals/directory.html
```

Chapter 20

searchplay

The searchplay tool will analyze a log file, detect the logs for search operations, and replay the search operations on the specified directory server.

The tool’s usage text in the following output shows the command-line syntax:

```
usage: searchplay [-d] [-s] [-w waittime]
                  [-p port] [-h hostname] [-f file] ...
```

When the -d option is specified, the tool will only display the searches found in the log file, but it will not perform the search operations. When the -s option is specified, the tool will generate statistics about the searches found in the log file.

Further documentation for this tool was not available at the time of publication. Please look for the latest version of this guide at the following URL:

```
http://docs.iplanet.com/docs/manuals/directory.html
```

Chapter 21

viewcore

The viewcore tool analyzes the contents of a core file to extract the debugging information. The core file must be generated by an executable that was compiled to produce the debugging information. The executable must also be given as input to the viewcore command.

NOTE The viewcore tool is specific to the Solaris operating environment. The command is available only in the installation on the Solaris platforms.

The viewcore command has the following syntax:

```
viewcore executableFile coreFile outputFile
```

Further documentation for this tool was not available at the time of publication. Please look for the latest version of this guide at the following URL:

```
http://docs.iplanet.com/docs/manuals/directory.html
```

Chapter 22

dbscan

The dbscan tool analyzes and extracts information from an iPlanet Directory Server database file. Database files use the .db2 and .db3 extensions in their filename, depending on the version of the iPlanet Directory Server.

The tool's usage text in the following output shows the command-line syntax:

```
dbscan - scan a db3 file and dump the contents
    -f <filename>   specify db3 file
    -i              dump as an index file
    -e              dump as an entry (id2entry) file
    -l <size>       max length of dumped id list (default 4096)
    -n              display idl lengths only (not contents)
    -G <n>          (when used with -n) only display index
                    entries with more than <n> ids
    -r              show libdb record numbers, too
    -k <key>        lookup only a specific key
```

Further documentation for this tool was not available at the time of publication. Please look for the latest version of this guide at the following URL:

```
http://docs.iplanet.com/docs/manuals/directory.html
```

Chapter 23

Security Tools

The Netscape Security Services (NSS) are a set of open source libraries and tools for implementing and deploying applications based on open standards for Internet security. The security tools help you perform diagnostics, manage certificates and keys, manage cryptography modules, and debug SSL- and TLS-based applications.

The NSS security tools are only introduced in this chapter. Each tool is given a brief description followed by a URL to more complete documentation on the Mozilla or iPlanet web sites. If you are unfamiliar with the concepts of Internet security, you should start with the overview of the NSS project at the following URL:

```
http://www.mozilla.org/projects/security/pki/nss/overview.html
```

The iPlanet Certificate Management System product relies on NSS to provide "a highly scalable, easily deployable certificate infrastructure for supporting encryption, authentication, tamper detection, and digital signatures in networked communications." You can use Certificate Management System to set up and manage your own public-key infrastructure or to deploy a public certification authority. For more information, see the following URLs:

```
www.iplanet.com/products/iplanet_certificate/home_2_1_1ad.html
docs.iplanet.com/docs/manuals/cms/42/adm_gide/contents.htm
```

This chapter contains the following sections:

- Overview of Security Standards
- Managing Certificates and Keys
- Other Utilities

Overview of Security Standards

This section gives an overview of all of the security standards implemented by the NSS libraries and tools. The following list is copied from the Mozilla web site (`http://www.mozilla.org/projects/security/pki/nss/overview.html`), and further links to the official standard documents may be found at the same location.

- SSL v2 and v3 - The Secure Sockets Layer (SSL) protocol allows mutual authentication between a client and server and the establishment of an authenticated and encrypted connection.
- TLS v1 (RFC 2246) - The Transport Layer Security (TLS) protocol from the IETF (Internet Engineering Task Force) will eventually supersede SSL while remaining backward-compatible with SSL implementations.
- PKCS #1 - RSA standard that governs implementation of public-key cryptography based on the RSA algorithm (invented by Rivest, Shamir, Aldeman). This algorithm is the basis for all PKCS (Public-Key Cryptography System) standards.
- PKCS #3 - RSA standard that governs implementation of Diffie-Hellman key agreement.
- PKCS #5 - RSA standard that governs password-based cryptography, for example to encrypt private keys for storage.
- PKCS #7 - RSA standard that governs the application of cryptography to data, such as for digital signatures.
- PKCS #8 - RSA standard that governs the storage and encryption of private keys.
- PKCS #9 - RSA standard that governs selected attribute types, including those used with PKCS #7, PKCS #8, and PKCS #10.
- PKCS #10 - RSA standard that governs the syntax for certificate requests.
- PKCS #11 - RSA standard that governs communication with cryptographic tokens (such as hardware accelerators and smart cards) and permits application independence from specific algorithms and implementations.
- PKCS #12 - RSA standard that governs the format used to store or transport private keys, certificates, and other secret material.
- S/MIME (RFC 2311 and RFC 2633) - IETF message specification based on the popular Multipurpose Internet Mail Extensions (MIME) standard and that provides a consistent way to send and receive signed and encrypted MIME data.

- X.509 v3 - International Telecommunication Union (ITU) standard that governs the format of certificates used for authentication in public-key cryptography.
- OCSP (RFC 2560) - The Online Certificate Status Protocol (OCSP) governs real-time confirmation of certificate validity.
- PKIX Certificate and CRL Profile (RFC 2459) - The first part of the four-part standard under development by the Public-Key Infrastructure (X.509) working group of the IETF (known as PKIX) for a public-key infrastructure for the Internet.
- AES, RSA, DSA, Triple DES, DES, Diffie-Hellman, RC2, RC4, SHA-1, MD2, MD5 - Common cryptographic algorithms used in public-key and symmetric-key cryptography.

Managing Certificates and Keys

The tools in this section store, retrieve and protect the keys and certificates on which encryption and identification rely. The following URL provides documentation about how keys and certificates are used to protect data and identity in secure communication:

`docs.iplanet.com/docs/manuals/security.html`

The security tools are located in *installDir*/`lib/nss/bin` of your iPlanet DSRK installation. In order to run them, you will need to add the following library locations to your `LD_LIBRARY_PATH` environment variable:

installDir/`lib/nss/lib:`*installDir*/`lib`

certutil

The `certutil` command manages certificate and key databases (`cert7.db` and `key3.db` files).

See the following web site for more information:

`www.mozilla.org/projects/security/pki/nss/tools/certutil.html`

cmsutil

The `cmsutil` command performs basic certificate management operations such as encrypting, decrypting, and signing messages.

See the following web site for more information:

```
www.mozilla.org/projects/security/pki/nss/tools/cmsutil.html
```

modutil

The `modutil` command manages the database of PKCS #11 modules (`secmod.db` files). It allows you to add new cryptography modules or modify the properties of existing modules, such as whether a module is the default provider of some cryptography service.

See the following web site for more information:

```
www.mozilla.org/projects/security/pki/nss/tools/modutil.html
```

pk12util

The `pk12util` command imports and exports both keys and certificates to and from their respective database and file formats defined by the PKCS #12 standard.

See the following web site for more information:

```
www.mozilla.org/projects/security/pki/nss/tools/pk12util.html
```

signtool

The `signtool` command creates signed jar archives containing files, code, or both. This command is fully documented in the Certificate Management System.

See the following web sites for more information:

```
docs.iplanet.com/docs/manuals/cms/42/adm_gide/app_sign.htm
```

signver

The `signver` tool verifies signatures on digitally-signed objects, such as the jar files signed with `signtool`.

See the following web site for more information:

```
docs.iplanet.com/docs/manuals/security/signver/signver.htm
```

ssltap

The ssltap tool can proxy requests for an SSL server and display the contents of the messages exchanged between the client and server. The tool does not decrypt data, but it shows the type of SSL message, for example clientHello or serverHello, and connection data such as the protocol version and cipher suite. The ssltap tool is used for debugging SSL messages in the Certificate Management System.

```
www.mozilla.org/projects/security/pki/nss/tools/ssltap.html
docs.iplanet.com/docs/manuals/cms/42/adm_gide/app_ssld.htm
```

Other Utilities

This section list the remaining, mostly undocumented commands provided with the iPlanet DSRK in the *installDir*/lib/nss/bin directory.

The following format conversion commands are documented in docs.iplanet.com/docs/manuals/cms/42/adm_gide/app_tool.htm:

- atob - Converts ASCII base-64 encoded data to binary base-64 encoded data.
- btoa - Converts binary base-64 encoded data to ASCII base-64 encoded data.

The remaining tools have only command-line documentation available by invoking the command without any arguments:

```
bltest
certcgi
checkcert
client
crlutil
derdump
digest
instinit
makepqg
newuser
ocspclnt
oidcalc
p7content
p7env
p7sign
p7verify
pp
rsaperf
sdrtest
selfserv
strsclnt
tstclnt
```

Chapter 24

Unsupported Utilities

This chapter gives a brief description for each of the Perl scripts provided in the *installDir*/unsupported/perl directory.

The unsupported utilities are small scripts that perform a variety of maintenance tasks. Some perform specific, programmed actions and will not work outside of their designated context. Others do not perform error checking and should be used only if you understand their behavior.

iPlanet makes no claim as to the suitability or correctness of these scripts. Use them at your own risk. The scripts are provided solely as examples of Perl shell programming to access a directory.

Table 24-1 Unsupported Utilities in *installDir*/unsupported/perl

Script Name	Intended Usage
adduser.pl	Add users to an LDAP directory.
changes2ldif.pl	Find an entry with a given changenumber.
corescoop.pl	Move mail server core files into a publishable area.
dn2ldif.pl	Read a set of DNs and produce the LDIF output.
export_auto_home.pl	Generate the equivalent of a /etc/auto.home file with data extracted from an LDAP directory.
export_mailgroups.pl	Generate the equivalent of a /etc/mailgroups.aliases file with data extracted from an from LDAP directory.
fixcopiedfrom.pl	Modify the copiedfrom attribute, which is sometimes needed to get replication working.
genaliases.pl	Generate the equivalent of a /etc/mlm-aliases/aliases.{intern,extern} file with data extracted from an LDAP directory.

Table 24-1 Unsupported Utilities in *installDir*/unsupported/perl *(Continued)*

Script Name	Intended Usage
genpasswd.pl	Generate the equivalent of a /etc/passwd file with data extracted from an LDAP directory.
import_aka.pl	Import the information of a SmartList mail alias into an LDAP directory.
import_auto_home.pl	Import the NIS auto.home map into an LDAP directory.
import_epage.pl	Import the EtherPage epage.users file into an LDAP directory.
inconsist.pl	Reports inconsistencies between LDAP and NIS data.
layoff.pl	Gives an example of how to automate certain system administration tasks.
ldap_mail.pl	Update mailRecipient information for a user.
ldap_migrate.pl	Move an LDAP entry from one specific server to another.
ldap_stress.pl	Perform a stress test with searches simulating Message Server load on an LDAP server.
ldappasswd.pl	Change the password of one or more users.
ldapstats.pl	Gather statistics from an LDAP server.
lfinger.pl	Perform an LDAP search with a command line that emulates the UNIX finger command.
mgroup.pl	Manage a mailing list or any other group from the command line.
migrate_user.pl	Move an entry from one subtree to another by creating the new entry, copying the attributes, and deleting the old entry.
modclass.pl	Add or delete one or more object classes from one or more entries.
normphones.pl	Normalize phone numbers and make sure they have the correct US area code.
qsearch.pl	Perform a quick and simple LDAP search for an entry.
rand_mods.pl	Modify or delete an attribute for one or more entries.
renattr.pl	Rename one or more attributes in entries matching the search criteria.
repstat.pl	Check the replication status on a Master and its replica.
restarter.pl	Restart a DS server under certain conditions.

Table 24-1 Unsupported Utilities in *installDir*/unsupported/perl *(Continued)*

Script Name	Intended Usage
rmduplicates.pl	Remove duplicate attribute values from all entries.
rmentry.pl	Remove one or more LDAP entries interactively.
tabdump.pl	Generate a tab-separated output of entries matching the search criteria.
termuser.pl	Similar to layoff.pl: gives an example of how to automate certain system administration tasks.
uidsynch.pl	Synchronize all UIDs with their mail attribute.
vrfyPO.pl	Check the consistency between the data on a mail server and in an LDAP directory.
vrfymail.pl	Verify that the mail and mailalternate attributes are not duplicates for any entry in a directory.
wits_dump.pl	Display a subset of attribute values for all entries in the directory.

Part 6

iPlanet LDAP Administrative Shell

Chapter 25

ilash Overview

The iPlanet LDAP Administrative Shell, called ilash, is an LDAP client that provides a scritable command-line interface to a directory. It provides commands for performing LDAP operations, navigating the directory, and importing and exporting data. It allows directory administrators to access configuration information and develop scripts for managing the directory.

The ilash tool is based on Tcl (Tool Command Language), and includes a complete Tcl interpreter, which has been extended to include commands for accessing the directory. The current version of ilash is based on Tcl 8.2.

The ilash tool provides an interface similar to that provided by a UNIX shell or Windows NT command prompt. Using ilash, you can navigate the directory hierarchy in much the same way as you can change the current directory in a file system. Similarly, directory locations in ilash can be specified as a relative or absolute DN in the same way that pathnames are specified.

This chapter contains the following sections:

- Starting the Shell
- The ilash Environment
- Summary of Commands
- Configuring ilash
- Scripting Example

Starting the Shell

This section describes how to launch the ilash shell and provides details about its command interpreter. The directory commands available in the shell and in ilash scripts are described in detail in Chapter 26, “ilash Command Reference.”

Syntax

The syntax of the `ilash` command line has three forms:

```
ilash -noconnect
ilash [-file TclScript] ...
ilash dbindOptions
```

Where:

- `-noconnect` launches the command interpreter without binding to a directory server. You should then use the dbind command to connect and bind to a server interactively.
- `-file` *TclScript* will interpret the given file as an `ilash` Tcl script. This script must also use the `dbind` command to establish a connection before being able to access the directory with other `ilash` commands.
- *dbindOptions* are the command-line options described in the next section. They are the same options as those of the `dbind` command, as the `ilash` tool attempts to establish a connection as soon as it is launched. You may omit any of the *dbindOptions* needed to connect and bind to a directory if they are defined in the configuration files (see "Configuring ilash," on page 214).

Options

The `ilash` command-line options are exactly those of the `dbind` command. Only those needed during startup are listed in the following table. The complete description of all options is given in "dbind," on page 231.

Table 25-1 Options for the `ilash` Tool

Option	Purpose
`-call` *URL-alias*	Specify the *URL* or *alias* name of the directory server target for the connection. Alias names are defined in the `ilash` configuration files (see "Configuring ilash," on page 214). The default server is: `ldap://localhost:389/`
`-user` *bindDN*	Specify a bind DN for accessing your directory, usually in double quotes (`""`) for the shell. If the bind DN and its password are omitted, the tool will use anonymous binding. The access permissions of the bind DN determine what entries are visible.
`-v2`	Perform an LDAP v2 bind. Use this options for servers that do not support LDAP v3. The default is to perform an LDAP v3 bind.

Table 25-1 Options for the `ilash` Tool *(Continued)*

Option	Purpose
`-noschema`	Specify that the ilash interpreter should not import the schema from the directory server. Some commands offer functionality that depends on the imported schema. By default, the `ilash` tool will import the schema when it binds to a directory.
`-nomove`	Specify that the initial location in the directory should be the root DSE. Otherwise, it will be the default naming context defined by the server.

CAUTION You should never specify the `-password` option on the command line because this exposes the *password* to other users through the list of processes (the `ps` command) on UNIX systems.

Specify the password in protected configuration files or give the password interactively when prompted by the `ilash` tool.

The initial binding and run-time behavior of the tool is fully customizable through configuration files, see "Configuration Files," on page 214.

The commands for interacting with the directory server are listed in "Summary of Commands," on page 212 and fully described in Chapter 26, "ilash Command Reference." While running `ilash`, you also have access to all Tcl commands that allow you to navigate the file system as well (see "The tclsh Shell," on page 210).

To exit `ilash`, type `quit` or use the `dunbind` command.

Directory Server Dependencies

When binding to a server, `ilash` will attempt to discover the capabilities of the server as published in the `supportedControl` and `supportedExtension` attributes in the root DSE of the server. Use the `dstatus -controls -extensions` command to view the supported controls and extensions that were discovered.

When possible, `ilash` will attempt to make use of these controls and extensions, and some command functionality will not work without them. However the `ilash` commands will not use a control or extension unless it is supported. They will return an error if they are unable to obtain a result by any other means.

Directory Schema

The ilash interpreter is able to detect, import and interpret the directory schema. It uses this information to format attributes for display and to generate templates for adding entries of any given object class.

By default, the schema is not imported, but the bind process will list the schema entries that were detected. Importing the schema is not required, the ilash tool will function normally without it. You may want to import the schema with the following command if you use the dadd command to compose new entries:

```
dschema "cn=schema" -import
```

You may also import any other sub-schema definitions as needed.

Parallel Operations

Many ilash commands, such as dshow, accept multiple targets. When possible, ilash will attempt to issue multiple operation requests in parallel to the server. By default, ilash will limit itself to 10 simultaneous requests to a server.

The degree of parallelism can be set by the maxconns parameter in the configuration files. Because simultaneous requests need separate connections, the ilash tool will rebind multiple times to achieve the parallelism. If sequential operation or single binding is required, the maxconns parameter should be set to 1.

The ilash Environment

The ilash interpreter allows you to conceptually navigate through the directory structure. Like a standard shell, it maintains a record of your current location in the directory and allows you to specify relative locations.

Navigation

The current location is an entry in the directory tree. The ilash interpreter always displays this entry’s DN in abbreviated form in the command prompt. The current location is always the default target entry for a command. As in a file system, the current location is abbreviated "." (a dot).

You may also specify an entry relative to the current location. The parent entry is abbreviated ".." (two dots), and children may be referred to by their RDNs (relative distinguished names). However, the parent notation may not be combined with any other: you must use absolute DNs to specify grandparents or siblings relative to the current location.

The following screen output shows simple navigation through the directory:

```
[People, Siroe,com]% dpwd
ou=People, dc=Siroe, dc=com
[People, Siroe,com]% dcd ..
[Siroe,com]% dshow "uid=bjensen, ou=People," -type cn
dn: uid=bjensen, ou=People, dc=Siroe,dc=com
cn:                   Barbara Jensen
cn:                   Babs Jensen
[siroe,com]% dls
2  ou=Groups,
3  ou=People,
4  ou=Special Users,
```

The Location Stack

Normally, the `ilash` interpreter does not remember previous locations as you move from one entry to another in the directory. However, it provides a location stack that allows you to store a location and move back to it at any time.

You store a location by pushing it onto the stack with the `-push` option of the `dmoveto`, `dcd`, `dlist`, `dls`, `dshow`, and `dshowentry` commands. You may push any number of locations onto the stack, but you will have to visit them all in reverse order to return to the desired location.

The `-pop` option of the `dmoveto` and `dcd` commands will change the current location to that of the last location pushed onto the stack.

The location stack can be used to create navigational shortcuts back to often visited locations. You may also use the location stack in scripts as a simple way to navigate through the entire directory.

‘The World’

A directory server may have several naming contexts that are entire directory trees. The root DSE is a special entry that represents the server itself and that lists the DN of each of its naming contexts. The `ilash` tool refers to the root DSE entry as ’The World.’ The attributes of this virtual entry are the server configuration parameters that may also be viewed with the `dstatus` command.

The root DSE may be referenced by an empty DN (""). You may change the location to the root DSE and view its contents, but it has no children. Also, the move -pop command on an empty stack will move to the root DSE.

There are other server configuration entries, for example cn=schema that contains schema definitions. These may be accessed through their DN as any other entry.

Sequences

Another navigational aid is called a sequence: it stores the results of directory operations to be used as targets for other commands. Both the dlist and the dsearch command return lists of DNs, all of which are stored in a sequence for later reference. The output from these commands shows the sequence number to the left of the DN for each result.

Sequence numbers may then be used as targets of other operations. You may also specify ranges of sequence numbers, for example 1-10, for commands that operate on multiple entries. The results of several commands may be appended to the same sequence, and duplicate entries will only appear once.

Sequences may be named and kept intact while performing other operations. Entries may be accessed by sequence name and number, for example myresults.1-10, or you may refer to all entries by giving the sequence name. For more information, see the reference for “dsequence,” on page 254.

All of these features make sequences useful in scripts for managing the output and performing complex operations.

The tclsh Shell

The ilash tool is built on top of the Tcl interpreter tclsh which means it has the full functionality of this command shell. Through the tclsh commands, you have full access to your file system and the resources of your machine.

This shell also offers useful functionality for interacting with ilash commands. For example, the history command of the tclsh shell can be used edit and replay previous commands. You may also write Tcl scripts that use ilash commands and execute them with the ilash interpreter. See “The .lashrc File,” on page 220 and “Scripting Example,” on page 220.

However, you do not need to know Tcl to use ilash. Chapter 26, “ilash Command Reference,” and the following section on quoting rules give you all the information you need to navigate through a directory and view its entries.

If you find bugs in the `ilash` scripts or customize them for added functionality, we encourage you to share your work with other users. Please post a message to the `iplanet.server.idsrk` public newsgroup with your ideas or your code.

Tcl Quoting

The `ilash` tool uses the quoting rules of the underlying `tclsh` shell. Like most shell interpreters, you must quote strings and parameters that contain spaces. However, you should be aware of the following rules for Tcl quoting:

- Both braces `{}` and double quotes `""` are used to enclose literal strings that contain spaces, for example:

  ```
  {ou=Human Resources,dc=Siroe,dc=com}
  "ou=Human Resources,dc=Siroe,dc=com"
  ```

- Within braces, commas that do not separate DN components must be escaped with a single backslash \ character:

  ```
  {o=Siroe.com\, SARL,c=fr}
  ```

- Within double-quoted strings, the same commas must be escaped with double backslash \\ characters:

  ```
  "o=Siroe.com\\, SARL,c=fr"
  ```

- The backslash is the escape character in Tcl, so pathnames on Windows NT will also require double backslashes. They will also require quoting if they contain spaces:

  ```
  "C:\\Program Files\\iPlanet"
  ```

  The Tcl interpreter also recognizes Windows NT pathnames containing forward slashes for which no escape character is necessary:

  ```
  "C:/Program Files/iPlanet"
  ```

Tcl Resources

The following sources provide more information for learning to use the `tclsh` shell and programming in the Tcl language.

- “Tcl Developer Site” - `tcl.activestate.com`
- “Tcl/Tk Documentation” - `tcl.activestate.com/doc/`
- “A Short Tcl Tutorial” - `www.cujo.com/tcl_tut.html`

- *Practical Programming in Tcl and Tk,* by Brent Welch.
- *Tcl and the Tk Toolkit*, by John Ousterhout.

Summary of Commands

The following sections give a summary of the `ilash` commands grouped by task categories. The tables also show synonyms for commands, which are simply alternate command names for the same functionality. All commands and their options are described in detail in Chapter 26, “ilash Command Reference.”

LDAP Commands

The LDAP commands are the `ilash` commands for standard directory operations.

Table 25-2 LDAP Command Equivalents of the `ilash` Tool

Command	Effect on the Directory
`dbind`	Perform an LDAP bind operation.
`drebind`	Alias for the `dbind -nomove -rebind` command.
`dextension`	Invoke an extension, given by its OID, on the server.
`dsearch`	Perform an LDAP search operation using a filter.
`dcompare`	Perform an LDAP compare operation on an attribute value.
`dadd`	Compose an entry in LDIF and perform an LDAP add operation.
`dmodify` `dmod`	Perform an LDAP modify operation to add, modify or remove attributes.
`dmodifyrdn` `dmodrdn`	Perform an LDAP modify-RDN (relative distinguished name) operation.
`drelocate`	Move entries from one subtree to another (“reparent”) or move entire subtrees.

`ddelete`
`drm`
`dbulkclean`
	Perform an LDAP delete operation with the option to delete subtrees.

`dunbind`
`dquit`
`quit`
	Close the connection to the current directory and optionally exit the `ilash` tool.

Directory Navigation

The navigation commands allow you to view the contents of a directory in the same way that standard shells or command interpreters allow you to navigate through a file system.

The navigational aids also include the commands for accessing the directory configuration information such as the schema and the supported controls.

Table 25-3 Directory Navigation Commands of the `ilash` Tool

Command	Effect on the ilash Environment

`dshowname`
`dshown`
`dpwd`
	Display the full DN of the current location or of the specified target object.

`dshowentry` `dshow`	List the attributes of the entry at the given location.
`dlist` `dls`	List the RDNs of all first-level children of a given entry.

`dmoveto`
`dmove`
`dcd`
	Change the current directory location of the `ilash` interpreter.

`dstatus` `dinfo`	Give information about the connection to the current directory server.
`dschema`	List the schema entries associated with an object or import schema definitions.
`dsequence` `dseq`	Manage the existing sequences and run commands using their entries.

`dhelp`
`help`
`?`
	Display the list of all `ilash` commands and their brief description.

Bulk Data Commands

There are two `ilash` commands that provide facilities for managing bulk data in the directory. The `dbulkclean` command is actually a synonym for `ddelete`.

Table 25-4 Bulk Data Commands of the `ilash` Tool

Command	Action
`dbulkload` `dload`	Import entries into the directory from either an LDIF text file or a CSV (comma-separated values) format file.
`dbulkdump` `ddump`	Export entries from the directory as LDIF text to a file or the standard output.

Configuring ilash

There are three files that control the operation of `ilash`:

- The system-wide configuration file.
- A user's configuration file.
- A user's `.lshrc` initialization script.

The configuration files allow you to set up global and personal defaults. The files contain configuration parameters to define aliases, set preferences and specify default command options. The initialization script contains `ilash` and Tcl commands that will be executed as the first commands of the `ilash` interpreter.

None of the files are mandatory, all functionality of the ilash commands is available through their command-line options.

Configuration Files

The system and user configuration files contain default information concerning remote servers, authentication and other general configuration options. The configuration files must have the following names:

- The system configuration is *installDir*`/etc/system.lashconfig.`
- A user’s configuration is the `.lashconfig` file in their home directory.

The *installDir* is the location of your iPlanet Directory Server Resource Kit installation, which is `/opt/iPlanet` by default.

These files contain simple parameter definitions that are loaded into the `ilash` interpreter. They may also contain comments on lines beginning with a hash (#). The exact syntax of these files is given in the following section, “Configuration Parameters.”

On startup, `ilash` will first read the system configuration file, if it exists, and then the user's configuration file, if it exits. Parameters and aliases that are redefined in the user's configuration file will override their definition in the system configuration file. User specifications and server specifications in a user’s configuration will take precedence over those of system configuration.

In general, information about servers and usage restrictions should be configured in the system configuration file. For example, a system administrator could set the system configuration file so that passwords are never sent openly during authentication. Certain security parameters such as password protection may not be overridden by users. Bind credentials and personal preferences for command defaults are more appropriate in the user's configuration file.

Both system and user configuration files might contain passwords in plain text and should be protected appropriately. For example, on UNIX platforms, you should make these files unreadable to others.

Configuration Parameters

The configuration files contain five types of parameters: global parameters, aliases, command defaults, server specifications, and user specifications.

Global Parameters

Most of the global parameters define a default value for bind operations. Each parameter is given as a pair on a separate line:

parameterName value

The *parameterName* is case insensitive, and the *value* should be quoted according the rules described in “Tcl Quoting,” on page 211. Some configuration parameters do not require a value, they take effect simply when they appear in a configuration file. The parameters are listed in the following table.

Table 25-5 Configuration File Parameters for the `ilash` Interpreter

Parameter	**Effect on the ilash Environment**
`defaultserver` *URL-alias*	Specify the URL or alias name of the default directory server for all connections. When giving an alias name, it must be defined later in the same file.

Table 25-5 Configuration File Parameters for the `ilash` Interpreter *(Continued)*

Parameter	Effect on the ilash Environment
`authentication` *method*	Define the default authentication method to use when binding. The recognized *methods* are `none`, `simple`, `md5` (CRAM-md5), `hd` (http digest-md5), and `tls` (transport layer security). You must specify this parameter with a value other than `none` to automatically bind with the `username` and `password` values.
`username` *DN*	Specify the full distinguished name to use for authentication. This username will be presented only if the authentication method is defined with a value of `simple`, `md5`, or `tls`.
`password` *password*	Give the password to be presented when authenticating using methods that require a password.
`protect-password`	Force binds to use encryption so that passwords will never be sent in the clear. The `dbind` command will refuse to perform any connection with simple or no authentication. This flag has no counterpart, so setting it in the system configuration will enforce the policy for all users.
`protect-data`	Force binds to use encryption and authentication so that passwords and data transferred during LDAP operations are secure. The `dbind` command will refuse to perform any connection with without TLS authentication. This flag has no counterpart, so setting it in the system configuration will enforce the policy for all users.
`use-tls`	Specify that the `dbind` command should establish a TLS connection by default. This flag has no counterpart, so setting it in the system configuration will enforce the policy for all users.
`tls-verify`	Force the `ilash` client to check the server's TLS credentials and abort the connection if the server cannot be verified. This flag has no counterpart, so setting it in the system configuration will enforce the policy for all users.
`ssl_key` *filename*	Specify the absolute path of the file containing a user’s SSL private key. This and all other SSL parameters may be set in the system configuration if all users share a common identity for secure operations.
`ssl_cert` *filename*	Specify the absolute path of the file containing the user's SSL certificate.
`ssl_cafile` *filename*	Specify the filename of the SSL CA certificate file for verification of the LDAP server's certificate.
`ssl_capath` *filename*	Specify the absolute path of the SSL CA certificate directory for verification of the LDAP server's certificate.

Table 25-5 Configuration File Parameters for the `ilash` Interpreter *(Continued)*

Parameter	Effect on the ilash Environment
`ldapversion` *version*	Specify the default LDAP version to use for connections. The *version* should be either `2` or `3`.
`local_dit` *DN*	Specify the default location in the directory from which navigation should begin after all connections.
`maxconns` *number*	Specify the maximum number of simultaneous operations to be issued to a server. When possible, the `ilash` tool will create multiple connections and perform operations in parallel.
`notype` *attributeName ...*	Give a list of attributes that should not be displayed unless explicitly requested.
`service` *LDAP-specificOptions ...*	Give a list default options for controlling LDAP operations. This parameter define global default options for all commands listed in “LDAP-Specific Options,” on page 224. These defaults may be overridden by individual command defaults.

DN Aliases

DN aliases define an abbreviation for a directory entry. Each alias is given on a separate line with the following format:

alias distinguishedName

The distinguished name should be the full DN of the entry, for example:

```
employees ou=People,dc=Siroe,dc=com
```

The DN alias may then be used as the target entry or target location for `ilash` commands. For example, the following command will display the DN of all entries that are immediate children of the `ou=People,dc=Siroe,dc=com` entry:

```
dlist employees
```

Command Defaults

All `ilash` commands may be given default parameters to override their built in defaults. A command default has the effect of specifying the given options every time the command is invoked. Each command default is given on a separate line with the following format:

commandSpecifier option ...

A *commandSpecifier* identifies the `ilash` command and its synonyms to which these options will apply. Options include any values that they require and should be enclosed in double quotes (`""`). There can be any number of options, for example:

```
search "-filter objectclass=*" "-norelative" "-type cn" "-show"
```

This command default will make all search results contain the full DN and the single `cn` attribute. By specifying a default filter, the `dsearch` command may be invoked without any options within the `ilash` interpreter.

The command defaults override the `notype` and `service` parameters that also define default option for most commands. They can in turn be overridden by any option that the user specifies on the command line.

The following table shows the command specifiers (in bold) and the `ilash` commands to which the default will apply.

add	dadd	**modifyrdn**	dmodifyrdn dmodrdn
bind	dbind	**moveto**	dmoveto dcd
bulkclean	dbulkclean	**search**	dsearch
bulkload	dbulkload	**showentry**	dshow dshowentry
compare	dcompare	**showname**	dshowname dpwd
delete	ddelete drm	**status**	dstatus
list	dlist dls	**unbind**	dunbind dquit quit
modify	dmodify dmod		

Server Specifications

A server specification contains default values that are specific to a connection to the given server. Whereas global parameters apply to any connection, the parameters in a server specification will take precedence only when binding and interacting with the named server. Server specifications have the following format:

```
server URL {
   alias name
   parameter  value
   ...
}
```

All of the lines within server specification are optional, although you should specify either an alias or local parameters. When the `alias` is defined, the *name* may then be used instead of the *URL* when invoking the `dbind` command. The *parameter value* pairs may be configuration parameters, DN aliases, or command defaults. For example:

```
server ldap://test.Siroe.com:5150/ {
    alias test
    authentication simple
    username "cn=directory manager"
    password testtest
    add "-template /usr/tools/template"
}
```

This server specification will simplify binding by allowing you to connect with the following command:

```
dbind -call test
```

The server alias may also be used in the global `defaultserver` parameter to further simplify the `dbind` command line.

User Specifications

A user specification contains default values for parameters when authenticating and later bound as the given DN. User defaults override server defaults and global defaults for the same parameter, DN alias name, or command specifier. User specifications have the following format in the configuration files:

```
user DN {
    parameter value
    ...
}
```

User specifications are more common in personal configuration files so that users may set up their preferences. For example:

```
user "uid=bjensen,ou=People,dc=Siroe,dc=com" {
    password zyG3dw13
    protect-password
    add "-draft /home/bj/.draft" "-template /usr/tools/template"
}
```

This specification gives a password to be used when binding as `bjensen` and also specifies that the password should never be sent in the clear. User specification are not cumulative with server specifications so the command default for the `dadd` command must redefine the `-template` option value, if desired.

The .lashrc File

If you wish to further initialize the environment of the `ilash` interpreter, you may create a `.lashrc` file in your home directory. This Tcl script file will be sourced after the tool has read the configuration files and before performing the initial bind.

Any valid Tcl commands may appear in the `.lashrc` file, for example setting variables or defining procedures. This file may also contain the following line:

```
set auto_noexec {}
```

The effect of this line is to prevent the `ilash` interpreter from invoking commands it doesn't recognize. By default, `ilash` relies on the Tcl shell program, `tclsh`, to invoke any unrecognized command as a system command. This default behavior allows you to access the file system and other resources through the usual commands.

Scripting Example

The bulk data commands of the `ilash` tool can be used interactively or automated through a simple script. The following code example gives a sample `ilash` script that bulk-loads data from a CSV data file into the directory location at `ou=People,dc=Siroe,dc=com`. This shows how to perform simple file manipulation, call `ilash` commands, and output the result.

To customize the file, you would set the base and ou variables to the directory location where the data is to be loaded, and the `csvdata` and `template` variables to the correct paths for the data and template files. This script uses the template and data file given as an example in "dbulkload," on page 235.

For further help with Tcl scripting, see the references in "Tcl Resources," on page 211.

Code Example 25-1 Sample `ilash` Script for Bulkloading

```
#!/opt/iPlanet/bin/idsrk50/ilash -file

# allow username or other bind flags to be specified
# on the command line
eval [concat dbind $argv]

set base     "dc=Siroe,dc=com"
set ou       "ou=People,"
set csvdata  "/usr/tools/data.csv"
set template "/usr/tools/template"
```

Code Example 25-1 Sample `ilash` Script for Bulkloading *(Continued)*

```
proc bulkload_ou {base ou csvdata template} {
    # move to the level where we want to create the ou
    puts "Moving to $base"
    dmoveto $base

    # create a draft for the ou
    set filename [file join [glob ~] .scriptdraft]
    set fd       [open $filename w]

    puts  $fd "objectclass: top"
    puts  $fd "objectclass: organizationalUnit"

    close $fd

    # add the ou entry and move there
    puts "Creating $ou"
    dadd $ou -draft $filename -noedit
    dmoveto $ou

    # perform the bulkload
    puts "Bulkloading entries into [dshowname -ufn]"
    puts [dbulkload -csvdata $csvdata -template $template
                    -overwrite]
}

puts "Script started at [clock format [clock seconds]]"
puts "-------------------------------------------------"

catch {bulkload_ou $BASE $OU $CSVDATA $TEMPLATE} result
puts $result

puts "-------------------------------------------------"
puts "Script ended at [clock format [clock seconds]]"
```

Chapter 26

ilash Command Reference

This chapter contains the detailed reference for each command of the ilash tool. Each reference gives the command-line syntax, the list of options and a description. In many cases, an example shows a typical usage of the command.

Some commands are synonyms of others: they are aliases to other commands that are fully referenced. Synonyms have identical command-line options all of the same functionality. For a quick overview of all ilash commands, see “Summary of Commands,” on page 212.

This chapter contains the following sections:

- Common Options
- Command Reference Section

Common Options

This section describes the options that are common to many commands. These options are not repeated in the syntax or description of each command reference, unless their meaning or usage is different in a particular command.

Online Help

All of the commands support the -help option that will list all possible options for that command, including the common options. For example:

```
[Siroe,com]% dshowname -help
dshowname   - show the name of an entry,
[<object>] [-ufn] [-ldap] [-help]
```

Full option names are given in the help message and used in descriptions and examples of this book. However, the `ilash` interpreter will recognize the shortest unique prefix of an option name. This abbreviation is sufficient to select that option on that tool. Prefixes for the same option may be different for each command due to ambiguities with other option names.

LDAP-Specific Options

The `ilash` commands that perform directory access operations all have a set options for specifying LDAP functionality. The LDAP-specific options include limits and controls that may apply to an LDAP request. The following commands support the LDAP-specific options:

```
dadd                            dmodify  dmod
dbulkdump  ddump                dmodifyrdn  dmodrdn
dbulkload  dload                dmoveto  dmove  dcd
dcompare                        drelocate
ddelete  drm  dbulkclean        dschema
dextension                      dsearch
dlist dls                       dshowentry  dshow
```

LDAP-specific options either have a default value or are not mandatory, meaning that they are not needed in most cases. When they are needed, it is best to specify them in the `service` configuration parameter so that they always apply to all directory access commands (see “Configuration Parameters,” on page 215).

Any of these options may still be given on the command line, to override either the default behavior or the configuration file settings.

Table 26-1 LDAP-Specific Options

Parameter	Purpose
`-timelimit` *seconds* `-notimelimit`	Set a time limit in *seconds* for the LDAP request to complete. The `-notimelimit` option is the default. These options determine how long the `ilash` client will wait for a response to its request. The directory server may impose its own time limit for operations in either case. A directory search that reaches the time limit may return incomplete results.

Table 26-1 LDAP-Specific Options *(Continued)*

Parameter	Purpose
`-sizelimit` *entries* `-nosizelimit`	Set the maximum number of *entries* that search requests will return. The `-nosizelimit` option is the default. The directory sever may also impose its own size limit for results. A directory search that reaches the size limit will return incomplete results.
`-attributesizelimit` *bytes* `-noattributesizelimit`	Set the maximum size in *bytes* for individual attributes in entries returned from a search operation. The size of an attribute includes its name and all of its values. The `-noattributesizelimit` option is the default. These options will have an effect only if the server supports the attribute sizelimit control (OID 1.3.6.1.4.1.1466.29539.1).
`-dereferencealias` `-dontdereferencealias`	Determine the action to take when encountering an alias in the target of an operation. By default, aliases will be dereferenced and their target will be returned. Commands will not notify the user when an alias is dereferenced.
`-chaining` `-nochaining`	Allow or prohibit the use of chaining. These options will have an effect only if the server supports the no-chaining control (OID 1.3.6.1.4.1.1466.29539.5). The default behavior is then determined by the server configuration.
`-refer` `-norefer`	Determine whether or not to follow referrals. The default behavior is determined by the server configuration.
`-usecopy` `-dontusecopy`	Allow or prohibit the use of a shadowed copy of the data. These options will have an effect only if the server supports the no-copy control (OID 1.3.6.1.4.1.1466.29539.2). The default behavior is determined by the server configuration.
`-managedsait`	Allow an administrator to operate on server configuration entries in the directory. This option can be used only when bound as the directory manager. It will take effect only if the server supports the manage-DSAIT control (OID 2.16.840.1.113730.3.4.2).

Display Options

The commands that display entries have common option for formatting their output. The display options control how entries are read from the directory and displayed in the output of the command. The following commands support display options:

```
    dsearch    dshowentry    dshow
```

Again, you may wish to set these options as defaults for the commands in a configuration file.

Table 26-2 Entry Display Options

Parameter	Purpose
`-type` *attribute ...*	Specify that only the listed attributes are read from the directory and displayed in the results. This option may also be used to explicitly request operational attributes such as the `subschemsubentry` attribute for schema references. The default behavior is that specified by the `-all` option below.
`-notype` *attribute ...*	Display all attributes except those listed. This option does not prevent the attributes from being read from the directory, it only filters them out of the display. The default behavior is that specified by the `-all` option below.
`-all`	This flag requests that all normal attributes of an entry be read from the directory and displayed in the output. This is the default behavior when the `-type` and `-notype` options are not used. However, operational attributes used by the directory server, such as the `subschemasubentry` attribute for schema references, are never displayed unless specifically requested using the `-type` option.
`-noall`	This option currently has no effect.
`-value` `-novalue`	These options determine whether or not attribute values are displayed. When `-novalue` is given, only attribute names are displayed under the DN for each entry, without any trailing punctuation. The `-value` option is the default behavior.
`-key` `-nokey`	These options determine whether or not attribute names, also known as keys, are displayed. When `-nokey` is given, only attribute values are displayed under the DN for each entry. The `-key` option is the default behavior.
`-show` `-noshow`	These options control the overall display of attributes under each DN in the output. The `-noshow` option will suppress all attribute display. With the `-show` option, attributes will be displayed according to the other display options above. `-show` is the default for the `dshowentry` command. `-noshow` is the default for the `dsearch` command.

The default behavior listed for each command option in this reference chapter may be modified by configuration files (see “Command Defaults,” on page 217). Options usually exist in pairs to allow either the command default or the configured default to be overridden. For example, the dshow command might be configured to not show attribute names. It is still possible to display the attribute names, but you must specify dshow -key.

Specifying Target Objects

Nearly every command requires you to specify the distinguished name of the entry that is the target of the command. Some commands will accept more than one entry as the target.

There are several methods available for specifying a target object:

- A fully qualified DN may specify a target anywhere in the directory.
- When an explicit target is not specified, the current directory position is the default target for all ilash commands.
- The ".." object refers to the parent of the current directory location, and "." refers to the current directory location.
- A relative DN specifies a target in subtree rooted at the current directory position. A relative DN contains a trailing comma (,), for example "uid=bjensen,ou=People," is valid when the current location is "dc=Siroe,dc=com".

  Relative DNs may not be combined with the parent ".." notation.
- An abbreviation defined in the user's .lashconfig file can be used as a target: the command will operate on the entry to which it refers.
- A valid sequence number will designate its corresponding entry in the current sequence as the target. Specify the sequence name and a dot separator (.) in front of the sequence to refer to an entry in another sequence. Ranges of sequence numbers may be used with commands that accept multiple targets such as dshow. These commands will also accept a mix of ranges and named sequence identifiers, for example:

  ```
  dshow 1-10 12 mysearch.3-7
  ```

Command Reference Section

?

Synonym for the `dhelp` command.

dadd

Compose an entry in LDIF and perform an LDAP add operation.

SYNTAX

```
dadd
    [object] [-objectclass objectclass ...]
    [-draft draftLocation] [-newdraft]
    [LDAP-specificOptions]
```

```
dadd
    [object] [-template templateFile]
    [-[no]edit]] [LDAP-specificOptions]
```

OPTIONS

object	Specify the DN of the entry to be added. The DN will be used in the new draft of an LDIF entry to which you add attributes and values before the command can add it to the directory. If you specify a relative DN, it will be expanded to its full value. The default value is the DN of the current location.
`-objectclass` *objectclass ...*	Specify the one or more object classes to which the new entry will belong. The tool will generate a new draft that, if the directory schema is loaded, will contain all parent object classes and all required and possible attributes. The default object class is `person`, which also implies `top`.
`-template` *templateFile*	Specify a file location that contains your template for creating new entries. This template file may contain any text you wish for creating a single entry, except for its DN. You should always specify the *object* option with the `-template` option. The `-objectclass` option is ignored when using a template file. The template file will be copied to the draft location, and the DN determined by the *object* option (or its default value) will be appended as the first line of the file. Then, you will have the opportunity to edit its contents before it is used to add a new entry. The template file is never modified.

`-draft` *draftLocation*

Specify the location of the draft file. This is where the tool will generate an LDIF text file containing the entry to add. The default location is the `.lashdraft` file in the user's home directory.

If a file exists at that location, the tool will ask you whether or not to use it as the draft. If you choose to use the existing draft file, the *object* and *objectclass* options will be ignored. If you choose not to use an existing draft file, if you specify a template file, or if you specify the `-newdraft` option, any file at the draft location will be overwritten.

`-newdraft`

Force the tool to generate a new draft file using the values of the *object* and *objectclass* options. Any file at the draft location will be overwritten without the choice to edit it instead. You must edit this new draft file to specify all necessary attribute values.

`-edit`
`-noedit`

Specify whether or not you want to edit the draft file interactively before it is used to add an entry. The default is the `-edit` option.

Use the `-noedit`, `-template`, and *object* options to add entries without any interaction. The template file and *object* must specify a complete and valid LDIF entry. This option is not compatible with `-newdraft`.

LDAP-specificOptions

See further options in "LDAP-Specific Options," on page 224.

DESCRIPTION

The `dadd` command creates an entry in LDIF, allows you to edit it, and then adds it to the directory. The credentials used to bind to the current directory must have permission to add the given entry.

There are two scenarios for creating an LDIF entry interactively:

- Specify a DN or relative DN along with object class names to generate a draft of the entry. If the directory schema has been imported with the `dbind -schema` option or the `dschema -import` command, the tool will automatically generate a draft with all object classes, their parent classes, and all possible attributes. The tool will open a text editor and you must complete the draft to create a valid entry in LDIF. When you close the editor, the tool will submit this entry in an LDAP add operation and rename the draft file with the suffix `.old`.
- Specify a DN and a template file that contains your draft for new LDIF entries. The tool will copy this file to the draft location and then open a text editor for you to complete the entry. When you close the editor, the tool will submit this entry as an LDAP add operation and rename the draft file with the suffix `.old`.

If an add operation fails, the draft will not be renamed, and you may run dadd again with the same command line. It will detect the draft and ask if you want to edit it. If yes, it will open the text editor, allowing you to modify and resubmit the entry. If not, it will create a new draft file using the DN and object class names or the template on the command line and overwrite the failed draft file.

The text editor is defined by the EDITOR variable in your environment. If this variable is not defined, you will be prompted for the name of a text editor application.

You may also use the dadd command to add an entry without interaction. Use the -noedit and -template options, specify the DN as the target *object*, and specify a complete valid LDIF entry, without the DN, in the template file.

EXAMPLES

The following command will create a draft entry and open it in an editor for you to complete:

```
[People,Siroe,com]% dadd "uid=bslater," -newdraft \
                         -objectclass organizationalPerson
```

The new draft will contain the parent object classes and all possible attributes for you to fill in (output edited for length):

```
dn: uid=bslater, ou=People, dc=Siroe, dc=com
objectClass: organizationalperson
objectClass: person
objectClass: top
#destinationIndicator:
#facsimileTelephoneNumber:
#internationaliSDNNumber:
#l:
#ou:
...
#sn:
#cn:
#description:
#seeAlso:
#telephoneNumber:
#userPassword:
```

The next example uses the following template file, which expressly has no DN:

```
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Ted Morris
sn: Morris
givenname: Ted
uid: tmorris
mail: tmorris@Siroe.com
telephonenumber: +1 555 058 4282
```

Because the template contains all required information, we may add the entry without editing the intermediate draft file:

```
[People,Siroe,com]% dadd "uid=tmorris," -template morris.ldif \
                          -noedit
```

SEE ALSO

`dbind -schema, dschema -import`

dbind

Perform an LDAP bind operation.

SYNTAX

```
dbind
   [-call URL-alias] [-noconnect] [-[no]move]
   [-v2|-v3] [-[no]schema] [-[no]rebind]
   [[-user] bindDN] [-password password]
   [-noauthentication] [-simple] [-hd]
   [-tls] [-tlsauth] [-[no]verify]
```

OPTIONS

`-call` *URL-alias*	Specify the URL or alias name of the directory server target for the connection. Alias names are defined in the `ilash` configuration files (see “Configuration Files,” on page 214). If the `defaultserver` option is not specified in the configuration files, the default server is `ldap://localhost/` on port 389.
`-rebind` `-norebind`	With the `-rebind` option, the `dbind` command will connect to the last server to which it was bound. It will also use the same connection parameters, unless they are overridden by options on this command line. The `-norebind` option is the default: the `dbind` command will connect to the default server. These options have no effect during the first connection of the `ilash` tool.

`-noconnect`	Prevents any binding so that the `dbind` command effectively does nothing. This option is used only when launching the `ilash` tool (see "Starting the Shell," on page 205).
`[-user]` *bindDN*	Specify the bind DN for authentication. The `-user` keyword is optional. The *bindDN* itself is optional even for strong authentication if the `username` parameter is defined in configuration files.
`-password` *password*	Specify the *password* for the given *username* DN. The password is optional because it may be specified in the configuration files for the given *username*. If this option is not specified and there is no password configured, the command will prompt the user for a password. This avoids ever having to store the password in a file.
`-noauthentication`	Perform anonymous binding, if allowed by the server. This option is the default if no *username* is not given.
`-simple`	Perform simple, non-encrypted and non-certificate based authentication with a username and a visible password. This is the default if a *username* is given.
`-md5`	Perform CRAM-MD5 SASL authentication to encrypt the password over the network.
`-hd`	Perform HTTP digest-MD5 authentication to encrypt the password over the network.
`-tlsauth`	Specify that the authentication process use the credentials presented by the user's certificate. The `-tlsauth` option will also set the `-tls` option for using encryption during the connection.
`-v2` `-v3`	Perform an LDAP v2 bind. Use this options for servers that do not support LDAP v3. The default is `-v3`: perform an LDAP v3 bind.
`-tls`	Specify that the connection should initiate a Start TLS operation with the server immediately after binding. This means that all communication to and from the server will be encrypted.
`-verify` `-noverify`	The `-verify` option will also set the `-tls` option. It will cause the `ilash` client to attempt to verify the certificate presented by the server for the TLS encryption. If the certificate cannot be verified, the connection will not be established. The default is `-noverify`: do not attempt to verify the server certificate.
`-schema` `-noschema`	Specify whether or not the `ilash` interpreter should import the directory's schema upon connecting to the server (see `dschema -import`). Some commands such as `dadd` offer functionality that relies on the imported schema. The default is `-schema`.

-move -nomove	The -move option specifies that the initial location in the directory should be the server's default naming context. The -nomove option specifies that the initial location should be the root DSE. The value of the local_dit option in the configuration files takes precedence over both, if it is defined. Otherwise, the default is -move.

DESCRIPTION

The dbind command connects the ilash tool to a directory server so that you can access its contents. On startup, the tool will bind to the server using this command and you may specify dbind options on the ilash command line.

You may also invoke the dbind command at any time to change servers or modify the bind parameters, such as the bind DN. Any existing connections will be automatically unbound first.

Many of the options such as the server URL and bind DN may be set as parameters in the configuration files to simplify the command line. The configuration files may also enforce an authentication mechanism that requires you to provide user and password credentials when using this command. The possible authentication mechanisms are:

- No authentication: no username or password is required.
- Simple: your password will be visible on the network; unfortunately, this is the only authentication supported by older LDAP v2 servers.
- CRAM-MD5: provides encryption of the password so that it is not exposed.
- HTTP digest-MD5: alternate encryption of the password so that it is not exposed.
- TLS authentication: provides a secure connection through certificates and encryption.

When using TLS authentication and encryption, you must define the SSL options such as ssl_key and ssl_cert in the configuration files.

EXAMPLES

The following example illustrates an anonymous bind to a server specified by its URL-format name:

```
[not bound]% dbind -call ldap://ldap.Siroe.com:398/
```

The second example shows how to initiate a secure connection with a server defined as SiroeSecure in the configuration files:

```
[not bound]% dbind -call SiroeSecure "cn=directory manager" \
                  -tls -verify

Enter password for "cn=directory manager":
```

SEE ALSO

`drebind`

dbulkclean

Synonym for the `ddelete` command.

dbulkdump

Export entries from the directory as LDIF text to a file or the standard output.

SYNTAX

```
dbulkdump
   [object ...] [-file filename] [-[no]header]
   [-[no]base] [-[no]below]
   [LDAP-specificOptions]
```

OPTIONS

object ...	Specify the root entry of the directory subtree to be exported. You may specify multiple objects: each of their subtrees will be output in the order specified.
`-file` *filename*	Specify the name of a file to contain the LDIF output. When this option is omitted, the command will use the standard output.
`-header` `-noheader`	Specify whether or not LDIF header information appears in the output. The header is a single line with a version identifier. The default is `-header`.
`-base` `-nobase`	Specify whether or not the root entries of the target subtrees are included in the output. With the `-nobase` option, the target objects on the command line will not appear in the output, only their subtrees will. The default is `-base`.
`-below` `-nobelow`	Specify whether or not subtree entries are included in the output. With the `-nobelow` option, only the target entries on the command line will appear in the output. The default is `-below`.
LDAP-specificOptions	See further options in “LDAP-Specific Options,” on page 224.

DESCRIPTION

The `dbulkdump` command writes entries in LDIF format to a file or to the standard output. Along with the `dbulkload`, this command provides the ability to back up your directory data or to move entries and entire subtrees to a different directory.

Export to the CSV (comma-separated value) format is not supported.

By default the tool will export all entries in the subtree rooted at the target object. The order of entries in the LDIF output will preserve the structure of the subtree hierarchy. Specifying multiple target objects or using the `-nobase` option may not preserve the structure, in which case the output cannot be bulk-loaded into another directory.

The output of the `dbulkdump` command is subject to the limit on the number of entries that may be retrieved at one time from the directory. This limit may be set on the client side by the `-sizelimit` option in "LDAP-Specific Options," on page 224. On the server side, the number is limited by the `nsslapd-sizelimit` attribute whose default is 2,000 entries. See "nsslapd-sizelimit (Size Limit)" in the *iPlanet Directory Server Command, Configuration and File Reference*. If either limit is reached, not all entries will be exported, and an error will be logged in the output.

SYNONYMS

`ddump`

SEE ALSO

`dbulkload`

dbulkload

Import entries into the directory from either an LDIF text file or a CSV (comma-separated values) format file.

SYNTAX

```
dbulkload
    [object] [-[no]overwrite]
    [-dereferencealias] [LDAP-specificOptions]
    -ldif filename
```

```
dbulkload
    [object] [-[no]overwrite]
    -csvdata filename -template filename
    [-rdn attributeName ...] [-[no]usefirst]
    [LDAP-specificOptions]
```

OPTIONS

`-ldif` *filename*

Define the path and filename of a file containing the LDIF entry records to import. This option in incompatible with all of the CSV-specific attributes.

`-overwrite`
`-nooverwrite`

Specify whether or not existing entries will be overwritten if the imported data contains the DN of an existing entry. The `-nooverwrite` option will report an error but not stop the import operation when attempting to add an entry that already exists. The default is `-overwrite`.

The `-nooverwrite` functionality involves fewer directory operations per entry. Therefore, if efficiency is a concern, specify this options when you know the imported DNs are unique. If you wish to replace entire subtrees, running `ddelete` then `dbulkload` with the `-nooverwrite` option will be faster.

`-dereferencealias`

Specify that the server should dereference any aliases in the imported entries. The default is `-dontdereferencealias`, unlike most other commands that have LDAP-specific options.

LDAP-specificOptions

The dbulkload command uses the LDAP-specific options for all of the search, delete, and add operations it performs internally. See “LDAP-Specific Options,” on page 224.

object

Specify a single directory location under which the data is to be loaded. The DN of this location will be used to create the DN of new entries specified by the CSV data. The default value is the current location in the directory.

`-csvdata` *filename*

Define the path and filename of a CSV file that contains attribute values for entries to be added. You must also specify a template file, and in most cases, you will need to specify `-rdn` attributes.

`-template` *filename*

Define the path and filename of a template file to translate CSV data into a valid entry. A template contains LDIF-like text and placeholders for attributes in the CSV data file.

`-rdn` *attributeName ...*

Specify any number of attributes that will form the relative distinguished name of the entries to be added from a CSV file. The values for these attributes will be used in the order given and appended to the front of the target `object` to create the DN of each entry. If no attribute names are specified using this option, `cn` is selected by default.

`-usefirst` `-nousefirst`	Specify whether or not to allow attributes given by the `-rdn` option to have multiple values in the new entry. Setting the `-usefirst` option allows RDN attributes to be multivalued and selects the first value defined in the CSV data or the template to figure in the RDN. The default is `-nousefirst`, which will report an error for an entry when one of its RDN attributes has more than one value.

DESCRIPTION

The `dbulkload` command loads multiple entries from a data file into the directory. The command has two distinct uses, one for each of the LDIF and CSV (comma-separated value) file types it can import:

- LDIF files should contain complete entry records, including the DN that determines the absolute location of new entries in the directory. The `dbulkdump` command and many other tools in the iPlanet Directory Server Resource Kit output LDIF files that can be imported into the directory.

  The dbulkload command does not support LDIF change records. It supports only LDIF attribute specification records that define all necessary attributes of an entry.

  LDIF files may contain a hierarchy of entries that form an entire subtree. Entries are imported in the order they appear, so parent entries must be given before their children. Also, the DNs of the new entries must fit into the existing directory hierarchy.

- The CSV format requires a template that determines how the values in the CSV data file are assigned to attributes. CSV is not a standard LDAP format, but many products such as spreadsheets can export columns or fields as comma-separated values.

  The target object given on the command line will be the parent of all new entries, meaning they may be loaded anywhere in the directory. However, all entries created from CSV data must have the same RDN structure, meaning that they must all be imported at the same level in the directory hierarchy.

Any errors in the format of an entry or when adding it to the directory will only prevent the entry in question from being loaded. The tool will output an error message but continue loading other entries. Finally, the credentials used to bind to the current directory must have permission to add all given entries.

EXAMPLE

The following example demonstrates bulk loading using CSV files. The CSV data file is a text file where each line is an entry with fields separated by commas:

```
Adam Alder, Alder, 555-0441, aa@Siroe.com, aa, {FILE}/h/adam.jpg
Bob Brown,  Brown,         , bb@Siroe.com, bb, {FILE}/h/bob.jpg
Carl Cade,   Cade, 555-0443,             , cc, {FILE}/h/carl.jpg
Dave Dibbs, Dibbs, 555-0445, dd@Siroe.com,   , {FILE}/h/dave.jpg
Eric Elmar,  Eric, 555-0447, ee@Siroe.com, ee, {FILE}/h/eric.jpg
```

A value containing whitespace, a comma, or a quote character should be surrounded by quotes. A quote is specified by two quotes in a row. For example:

```
employee,"Fred ""Ford""","3rd floor, room 12",0449
```

The template file contains an LDIF entry with placeholders for field values and no DN. The DN is generated from the target object and the `-rdn` option specified on the command line. The placeholders are replaced with their corresponding field number. The following is a sample template for the CSV data above:

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
sn: $2
cn: $1
telephoneNumber: $3
postalAddress: Siroe.com $ P.O. Box 555 $ Santa Clara, CA 94303
mail: $4
uid: $5
jpegPhoto: $6
```

Placeholders must appear as attribute values at the end of a template line and must refer to a valid field number. However, field numbers may be repeated and not all fields need to be used in the template.

The following examples use the CSV sample data and template files above:

```
[Siroe,com]% dbulkload "ou=People," -csvdata path/data.csv \
                       -template path/template -rdn cn uid
```

```
Added cn=Adam Alder + uid=aa, ou=People, dc=Siroe, dc=com
Added cn=Bob Brown + uid=bb, ou=People, dc=Siroe, dc=com
Added cn=Carl Cade + uid=cc, ou=People, dc=Siroe, dc=com
path/data.csv:4 error processing record: entry doesn’t have the
rdn attribute uid
Added cn=Eric Elmar + uid=ee, ou=People, dc=Siroe, dc=com
```

The “Scripting Example,” on page 220 also uses the `dbulkload` command to automate a bulk loading operation.

SYNONYMS

`dload`

SEE ALSO

`dbulkdump`

dcd

Synonym for the `dmoveto` command.

dcompare

Perform an LDAP compare operation on an attribute value.

SYNTAX

```
dcompare
    [object] -attribute name=value
    [-[no]print] [LDAP-specificOptions]
```

OPTIONS

object	Specify the DN of the entry in which to compare the given attribute value. The default is the entry at the current location.
`-print` `-noprint`	Specify the format of the output: `-print` (the default) will return the value `TRUE` or `FALSE.` `-noprint` will return the value `1` or `0`, respectively.
`-attribute` *name*=*value*	Specify the *name* of the attribute to verify and the *value* to which it should be compared. Multivalued attributes will compare positively if any one of their values equals the given *value*.
LDAP-specificOptions	See further options in “LDAP-Specific Options,” on page 224.

DESCRIPTION

The `dcompare` command checks to see whether or not the target entry holds a particular attribute value. It returns `TRUE` if the entry contains an attribute with the given name and value, `FALSE` otherwise. Use the `-noprint` option to return a numeric value of `1` or `0`, respectively, which is easier to test programmatically, as in in a script.

The LDAP-specific options are used as parameters to the internal directory access operation needed to read the attribute value.

ddelete

Perform an LDAP delete operation with the option to delete subtrees.

SYNTAX

```
ddelete
   [object ...] [-base] [-below] [-subtree]
   [-[no]prompt] [LDAP-specificOptions]
```

OPTIONS

object ...	Specify the target objects to be deleted. Unless you specify either the `-below` or `-subtree` option, the target must be a leaf entry.
`-base`	Specify that only the target objects should be deleted. This is the default behavior.
`-below`	Specify that all entries in the subtree below each target object should be deleted. The target entries are not deleted.
`-subtree`	Specify that the entire subtree rooted at each target object, including the targets themselves, should be deleted.
`-prompt` `-noprompt`	Specify whether or not the command should ask the user for confirmation before performing the delete operation The default is `-prompt`.
LDAP-specificOptions	See further options in “LDAP-Specific Options,” on page 224.

DESCRIPTION

The `ddelete` command removes entries from the directory, with the option to delete entire subtrees. When deleting a single entry, it must not have any children. When deleting a subtree, you specify a non-leaf entry and you have the choice of deleting only those entries below it. In all cases, the credentials used to bind to the current directory must have permission to delete entries.

The command will use the subtree delete control (OID 1.2.840.113556.1.4.805) when the server supports it. When the control is not available, the command will delete subtrees by recursively descending the directory tree to remove entries.

SYNONYMS

`dbulkclean, drm`

ddump

Synonym for the `dbulkdump` command.

dextension

Invoke an extension, given by its OID, on the server.

SYNTAX

```
dextension
   [-oid OID] [-data data] [-[no]force]
   [-stop] [-reset]
   [LDAP-specificOptions]
```

OPTIONS

`-oid` *OID*	Specify the OID (object identifier) of the extension in dotted decimal format, for example: `1.2.840.113556.1.4.805` .
`-data` *data*	Give additional *data* as a string that will be sent with the extension request. Some extensions require additional parameters that can be entered here.
`-stop`	This option is a shortcut for invoking a shutdown extension (OID 1.3.6.1.4.1.1466.20001) recognized by some servers.
`-reset`	This option is a shortcut for invoking a reset extension (OID 1.3.6.1.4.1.1466.20002) recognized by some servers.
`-force` `-noforce`	The `-force` option specifies that the extension should be invoked without checking whether the server supports it. The default is `-noforce`.
LDAP-specificOptions	See further options in "LDAP-Specific Options," on page 224.

DESCRIPTION

The `dextension` command invokes extensions on the directory server. Extensions are server-dependent operations that may perform non-LDAP operations or modify the functionality of the server. The tool will check whether or not the server supports the extension before sending it.

The directory server may send a reply to the extension request, in which case the `dextension` command will display the OID and data contained in the response.

dhelp

Display the list of all `ilash` commands and their brief description. This list does not include synonyms for commands.

SYNONYMS

```
help, ?
```

dinfo

Synonym for the `dstatus` command.

dlist

List the RDNs of all first-level children of a given entry.

SYNTAX

```
dlist
   [object] [-[no]move] [-[no]push]
   [-sequence name] [LDAP-specificOptions]
```

OPTIONS

object	Specify a single target object. The default is the current location.
`-move` `-nomove`	Specify whether or not the command should also change the current location to that of the target *object*. The `-move` option is not compatible with `-push`. The default is `-nomove`.
`-push` `-nopush`	Specify whether or not the command should also push the current location onto the location stack and change to the location of the target *object*. The `-push` option is overridden by `-move` and `-nomove` when they occur after it on the command line. The default is `-nopush`.
`-sequence` *name*	Save the DNs of entries from the output under the given sequence *name* and do not modify the current sequence. Sequence names may contain only alphabetical characters. If the sequence does not exist, it will be created. If the sequence exists, new DNs will be appended to the end of it. When this option is omitted, the DNs are appended to the current sequence.
LDAP-specificOptions	See further options in “LDAP-Specific Options,” on page 224.

DESCRIPTION

The dlist command searches the directory and displays the list of all entries that are immediate children of the target object. The output of the command is a list of RDNs that are relative to the target object, not to the current location.

In the output, the number to the left of each RDNs is its number in the sequence. If the corresponding DN already exists in the current or the named sequence, the numbers may not begin at 1 or even be numbered sequentially.

The number of RDNs returned can be limited by using the -sizelimit option described in “LDAP-Specific Options,” on page 224.

This command may be considered the equivalent of the ls or dir command, although it doesn’t support filtering on the DN. To search for specific DNs, you must use the dsearch command. To display the contents of an entry, use the dshowentry command.

SYNONYMS

dls

SEE ALSO

dsearch, dsequence, dshowentry, dmoveto

dload

Synonym for the dbulkload command.

dls

Synonym for the dlist command.

dmod

Synonym for the dmodify command.

dmodify

Perform an LDAP modify operation to add, modify or remove attributes.

SYNTAX

```
dmodify
    [object ...] [-draft draft [-[no]edit]] [-newdraft]
    [LDAP-specificOptions]
```

```
dmodify
    [object ...] [-add attribute=value] ...
    [-remove attribute=value] ... [-remove attribute] ...
    [LDAP-specificOptions]
```

OPTIONS

object ...	Specify the target objects to modify. The default is the current location. When using a draft file, you will be able to edit the target entries as LDIF in a text editor. When adding or removing only attributes, all target entries will be modified in same way.
`-draft` *draftLocation*	Specify the location of the draft file. This is where the tool will generate an LDIF text file containing the entries to modify. The default location is the `.lashdraft` file in the user's home directory.
`-edit` `-noedit`	Specify whether or not you want to edit the draft file interactively before it is used to modify entries. The default is `-edit`.
`-newdraft`	If a file exists at the draft location, the tool will ask you whether or not to use it as the draft. Use this option to suppress the query and force the tool to generate a new draft file from the target entries. Any file at the draft location will be overwritten. This option is overridden by the `-noedit` option.
`-add` *attribute*=*value*	Specify the name and initial value of an attribute to add. If the attribute is already present in the target, it becomes a multivalued attribute. This option may be repeated to add multiple attributes or multiple values to the same attribute.
`-remove` *attribute*=*value* `-remove` *attribute*	Specify the name of an attribute to remove. If you also specify a value, the attribute is assumed to be multivalued and only that value will be removed. When you specify only the attribute name, all values will be removed. This option may be repeated to remove multiple attributes or multiple values.
LDAP-specificOptions	See further options in "LDAP-Specific Options," on page 224.

DESCRIPTION

The `dmodify` command has two usage scenarios:

- Interactive: edit the target entries in an LDIF text file to add, modify, or remove attributes or to add and remove entire entries.
- Batch: specify on the command line the attribute values to add or remove in order to modify attributes on all target entries in the same way.

To modify entries interactively, the command generates the LDIF text for all target entries in a draft file and opens it in a text editor. You may then edit this file to modify attribute values, remove attributes or add new ones. You may also add or remove entire entries. The tool will then compare this file with the existing entries and perform all necessary LDAP operations to update the directory.

When you close the editor, the tool will display the number of added, modified, and deleted entries and ask for confirmation before committing all changes. If you change your mind, the directory will not be modified. Once you have edited the LDIF entries in the draft file you may:

- If you did not commit the changes and you now wish to do so, run the same command again with the `-noedit` flag. This will commit all changes in the current draft file with no further interaction.
- Whether or not you committed any changes, run the same command again and use the existing draft file. You will be able to make further modifications before committing.
- Specify the `-newdraft` option to ignore any existing draft and read the target entries again for editing.

The text editor is defined by the `EDITOR` variable in your environment. If this variable is not defined, you will be prompted for the name of a text editor application.

In the batch scenario, you specify modifications on the command line and the command performs them on all target entries. By using sequence numbers to specify targets, you may add, modify or delete an attribute across the entire directory. You may specify any number of attribute modifications on the command line, allowing you to perform global updates.

In both scenarios, the credentials used to bind to the current directory must have permission to modify all given entries. Finally, the server may also reject modifications that do not follow the schema of the directory.

SYNONYMS

```
dmod
```

dmodifyrdn

Perform an LDAP modify-RDN (relative distinguished name) operation.

SYNTAX

```
dmodifyrdn
    [object] -name name=value
    [-[no]delete]
    [LDAP-specificOptions]
```

OPTIONS

object	Specify a single target object. The default is the current location.
`-name` *name*=*value*	Define the new RDN of the target entry. The named attribute and its value will also be added to the entry.
`-delete` `-nodelete`	Specify whether or not the attribute and value corresponding to the old RDN component should be removed from the entry. The default is `-delete`.
LDAP-specificOptions	See further options in "LDAP-Specific Options," on page 224.

DESCRIPTION

The `dmodifyrdn` command modifies the RDN of an entry. The effect of this command is to rename the target entry by replacing the first component in its distinguished name. Use the `-nodelete` option if you wish to keep the old RDN as an attribute in the entry.

Most directory servers allow this operation on leaf entries only. Modifying the RDN of a non-leaf entry is equivalent to renaming the entire subtree below that entry. Use the `drelocate` command to modify entire subtrees.

After modifying an RDN, any sequence number referring to the target entry will still refer to the old RDN. If you use this old sequence number, the old RDN will no longer be recognized by the directory.

SYNONYMS

`dmodrdn`

SEE ALSO

`drelocate`

dmodrdn

Synonym for the `dmodifyrdn` command.

dmove

Synonym for the `dmoveto` command.

dmoveto

Change the current directory location of the `ilash` interpreter.

SYNTAX

```
dmoveto
    object
    [-[no]pwd] [-[no]pop] [-[no]push] [-[no]check]
    [LDAP-specificOptions]
```

OPTIONS

object	Specify a single target object to become the new location. If you omit the target object, the default value is the current location. Changing to the current location has no effect unless you use the `-push` option to place it on the stack. Specifying a target has no effect when using the `-pop` option.
`-pwd` `-nopwd`	Specify whether or not to display the DN of the new location. When using either `-push` or `-pop`, the `-pwd` option will also display the new contents of the location stack beneath the current location. The default is `-nopwd`, which will suppress all output. Regardless of any command options, the current location is always displayed in abbreviated form in the `ilash` command prompt.
`-pop` `-nopop`	The `-pop` option changes the current location to that of the DN at the top of the location stack and removes that DN from the stack. The `-pop` option will override all other options except when `-nopop` occurs after it on the command line.
`-push` `-nopush`	The `-push` option will place the current location on the location stack before changing to the new location. This option is not compatible with `-pop`. The default behavior is `-nopush`: the location stack is not modified, only the current location changes.

`-check` `-nocheck`	Specify whether or not to check the validity of the target location before making it the current location. The default is `-check`. This option has no effect when using the `-pop` option: locations popped from the stack are not checked.
LDAP-specificOptions	See further options in "LDAP-Specific Options," on page 224.

DESCRIPTION

The `dmoveto` command changes the current directory location to the given target DN, much like the `cd` command of a standard shell. You may also push the old location onto the location stack.

The `-pop` option will return to the location specified by the DN taken from the top of the location stack. Specifying an empty string (`""`) as the target object or using the `-pop` option when the location stack is empty will change to 'The World.' This location is the root DSE, a virtual entry representing the directory server itself (see "'The World'," on page 209).

When you specify a target DN, the tool will check the validity of the new location with the server. The LDAP-specific options are used only during this directory access. It will not change the current location if the target DN is not valid. If you use the `-nocheck` option and specify an invalid target location, all further commands that rely on the current location will return errors.

The current position can also be changed with the `-move` option of the `dlist` and `dshowentry` commands.

SYNONYMS

`dmove, dcd`

SEE ALSO

`dlist, dshowentry`

dpwd

Synonym for the `dshowname` command.

dquit

Alias for the `dunbind -quit` command.

drebind

Alias for the `dbind -nomove -rebind` command.

drelocate

Move entries from one subtree to another ("reparent") or move entire subtrees.

SYNTAX

```
drelocate
    [object ...] -parent object
    [-[no]copy]
```

```
drelocate
    [object] -target object
    [-[no]copy]
```

OPTIONS

object ...	Specify the objects to be moved. Only one object may be specified when using the `-target` option.
`-parent` *object*	Specify the parent entry under which all objects should be moved. The parent object must be the DN, relative DN or sequence number of an existing entry.
`-target` *object*	Specify the new DN of the object to be moved. The object to be moved will effectively be rename to this target DN. Therefore, the DN must not designate an existing entry, but its parent, given by the second DN component, must exist.
`-copy` `-nocopy`	Specify whether or not the `drelocate` command should perform the operation internally or call the LDAP rename operation. The default is `-nocopy`, which calls the LDAP rename operation. If the server is unwilling to perform the rename operation, use the `-copy` option to reparent the entries or move a subtree. The tool will do this internally by retrieving the entries to be moved, editing their RDN, adding them as new entries under the given parent or with the given target DN, and if all goes well, deleting the original entries.

DESCRIPTION

The drelocate command performs either the LDAP rename operation with the "newparent" option or adds and deletes entries to perform the equivalent operation. Many servers have limitations on relocating entries such as allowing only leaf entries to be moved and allowing them to be moved only within their original naming context.

Some servers may not even support the rename operation, in which case you must use the -copy option to perform the operation by copying entries. Most servers will not allow you to move subtrees with the rename operation, so use -copy as well. Even with this option, the drelocate command has its own limitations, and not all subtree configurations can be moved.

In all cases, the tool preserves all original entries if an operation can not be completed correctly, either by the server or by the -copy option.

drm

Synonym for the ddelete command.

dschema

List the schema entries associated with an object or import schema definitions.

SYNTAX

```
dschema
    [object ...] [-sequence name] [LDAP-specificOptions]
```

```
dschema
    [object ...] [-[no]import] [-[no]overwrite]
    [LDAP-specificOptions]
```

OPTIONS

object ...	Specify all objects for which to list the associated schema entries. When using the -import option, specify the subschema entries to import. The default is the current location.
-import -noimport	Specify whether or not the target objects are schema entries with definitions to import. When -import is used, all object class and attribute definitions in the target entries will be loaded into the ilash interpreter. The default is -noimport.

`-overwrite` `-nooverwrite`	Specify whether or not existing definitions should be overwritten when importing new definitions with the same OID. The default is `-nooverwrite`. These options are meaningful only when `-import` is specified.
`-sequence` *name*	Save the DNs of schema entries in the output under the given sequence *name* and do not modify the current sequence. Sequence names may contain only alphabetical characters. If the sequence does not exist, it will be created. If the sequence exists, new DNs will be appended to the end of it. When this option is omitted, the DNs are added to the current sequence. This option is ignored when `-import` is specified.
LDAP-specificOptions	See further options in “LDAP-Specific Options,” on page 224.

DESCRIPTION

The `dschema` command handles the special attribute values used to reference and define the schema in the directory. There are two forms to this command:

- The default behavior is to display the DN of schema entries that are referenced by the target entries. To view these schema definitions, use the `dshowentry` command on the DNs of schema entries in the output.

  Schema entries are referenced by the values of the `subschemaSubentry` attribute in the target entries. This attribute is available only in LDAPv3 directories, and `dschema` will report an error on LDAPv2 directories.

- When you specify the `-import` option, the `dschema` command will import any schema definitions found in the target objects. The target must be an entry of the objectclass `subschema`, whose `ldapSchemas`, `objectClasses`, `attributeTypes`, `ldapSyntaxes`, and `matchingRules` attributes define the schema.

  Importing the schema entries into the `ilash` interpreter allows some commands to offer functionality that relies on the directory schema. For example, the `dadd -objectclass` command uses the schema to generate a template file containing all parent classes and attributes of the given object class.

The LDAP-specific options are used in both cases when the command accesses the directory to read attribute values from the given entries.

SEE ALSO

`dbind -schema`

dsearch

Perform an LDAP search operation using a filter.

SYNTAX

```
dsearch
    [object]
    [-baseobject | -onelevel | -subtree]
    [-filter filter] [-[no]relative]
    [-matchedvaluesonly]
    [-[no]searchaliases] [-sequence name]
    [displayOptions] [LDAP-specificOptions]
```

OPTIONS

object	Specify the DN of the base entry for the search. The default is the current location.

`-baseobject`
`-onelevel`
`-subtree`
	Specify the scope of the search: either the base entry itself, only the immediate children of the base entry, or the entire subtree rooted at the base entry. The three scope options are mutually exclusive. The default is `-subtree`.

`-filter` *filter*	Specify an RFC 2254-compliant LDAP search filter, usually in double quotes (`""`) for the interpreter (see “LDAP Search Filters” in Appendix B of the *iPlanet Directory Server Administrator’s Guide*).
`-relative` `-norelative`	Specify whether or not the DNs in the output are shown in their relative or full format. Relative DNs are relative to the base entry of the search, not to the current location. The default is `-relative`.
`-searchaliases` `-nosearchaliases`	Specify whether or not aliases in the search scope are dereferenced before applying the filter. The default is `-nosearchaliases`. These options are different from the `-[dont]dereference` options listed in “LDAP-Specific Options,” on page 224. The latter apply only to aliases encountered when determining the base entry given by the target object, as opposed to the entries being searched in the scope.
`-matchedvaluesonly`	Specify that only attributes being matched by the search filter should be returned in matching entries. This functionality depends on a control that the directory server must support. By default, this option is not active.

-sequence *name*	Save the DNs of entries from the output under the given sequence *name* and do not modify the current sequence. Sequence names may contain only alphabetical characters. If the sequence does not exist, it will be created. If the sequence exists, new DNs will be appended to the end of it. When this option is omitted, the DNs are appended to the current sequence.
displayOptions	See other options in “Display Options,” on page 225.
LDAP-specificOptions	See further options in “LDAP-Specific Options,” on page 224.

DESCRIPTION

The `dsearch` command searches the directory for entries whose attributes match the filter in the given scope under the base DN. By default, only the DN or the relative DN of matching entries are displayed. Use `-show` along with the other options listed in “Display Options,” on page 225 to view selected attributes.

EXAMPLE

The following search returns the full name and telephone number of all entries with a last name (surname `sn`) that starts with `J`. It stores the reference to the corresponding DNs in the sequence also named `J`.

```
[The World]% dsearch "dc=Siroe,dc=com" -filter "sn=J*" \
                     -show -type cn telephonenumber \
                     -sequence J

J.1  uid=bjablons, ou=People,
cn:                    Barbara Jablonski
telephoneNumber:       +1 408 555 8815

J.2  uid=ejohnson, ou=People,
cn:                    Emanuel Johnson
telephoneNumber:       +1 408 555 3287

J.3  uid=kjensen, ou=People,
cn:                    Kurt Jensen
telephoneNumber:       +1 408 555 6127

J.4  uid=bjensen, ou=People,
cn:                    Barbara Jensen
cn:                    Babs Jensen
telephoneNumber:       +1 408 555 1862
```

dseq

Synonym for the `dsequence` command.

dsequence

Manage the existing sequences and run commands using their entries.

SYNTAX

```
dsequence
```

```
dsequence
   [-sequence] name
```

```
dsequence
   [[-sequence] name]
   [-reset] [-status] [-all]
```

```
dsequence
   [[-sequence] name]
   [-iterate TclCode]
```

OPTIONS

`[-sequence]` *name*	The name of the target sequence for the chosen option. The default is the current sequence.
`-all`	Specify that the `-reset` or `-status` option apply to all sequences. This option is not compatible with `-iterate`.
`-reset`	Remove all entries from the specified sequence. After a sequence is reset, referring to any of its entries will return an error. However, the name may be reused to create another sequence.
`-status`	List the number of entries, if any, in the specified list.
`-iterate` *TclCode*	Interpret the *TclCode* in the context of every entry in the list. The command will set the current location to each of the entry locations in turn and interpret the *TclCode* in that context.

DESCRIPTION

The `dsequence` command manages sequences and their contents, allowing you to swap the current sequence for another. The current sequence is the one to which `dlist`, `dschema` and `dsearch` results will be appended by default. You may refer to entries in the current sequence simply by giving their number on a command line. Entries are not duplicated in any one sequence, so an entry will always have the same number.

When the `ilash` interpreter is launched, the name of the current sequence is `default`. The `dlist`, `dschema` and `dsearch` commands may create new sequences with their `-sequence` *name* option. Sequence names may contain only alphabetical characters. The command:

```
dsequence [-sequence] name
```

will make the named sequence the current one. If the name sequence does not exit, it will be created empty and still become the current sequence. The command by itself will display the name of the current sequence:

```
[Siroe,com]% dsequence
default
[Siroe,com]% dsequence mysearch
[Siroe,com]% dsequence
mysearch
```

The contents of the previously current sequence are still available under its name. You may refer to any sequence contents at any time with the following notation:

sequenceName`.`*entryNumber*

Use the `-status` option to view the number of entries in each sequence. Sequences without any entries are not listed by this option. Use the `-reset` option to delete a sequence:

```
[Siroe,com]% dsequence -status -all
47    entries in sequence <mysearch>
34    entries in sequence <default>
4     entries in sequence <J>

[Siroe,com]% dsequence -reset default
[Siroe,com]% dsequence -status -all
47    entries in sequence <mysearch>
4     entries in sequence <J>
```

Use the `-iterate` option to run the given Tcl code in the context of each entry. This command will change the current location to each of the entries in the sequence and interpret the code. The Tcl code may be a simple command or it may be a complete script that uses Tcl commands, `ilash` commands, or both.

SYNONYMS

`dseq`

dshow

Synonym for the `dshowentry` command.

dshowentry

List the attributes of the entry at the given location.

SYNTAX

```
dshowentry
    [object ...] [-[no]name]
    [-[no]move] [-[no]push]
    [displayOptions] [LDAP-specificOptions]
```

OPTIONS

object ...	Specify one or more objects to display. The default is the entry at the current location.
`-name` `-noname`	Specify whether or not to display the DN (distinguished name) of the entry before its contents. The default is `-name`.
`-move` `-nomove`	Specify whether or not the command should also change the current location to that of the target *object*. The `-nomove` option overrides `-move` and `-push` when it occurs after them on the command line. The default behavior is `-nomove`.
`-push` `-nopush`	Specify whether or not the command should also push the current location onto the location stack and change to the location of the target *object*. When showing multiple entries, they will be pushed onto the stack in the order given. The `-push` option is overridden by `-move` and `-nomove` when they occur after it on the command line. The default is `-nopush`.
displayOptions	See other options in “Display Options,” on page 225.
LDAP-specificOptions	See further options in “LDAP-Specific Options,” on page 224.

DESCRIPTION

The `dshowentry` command will show the attributes contained in the given entry and optionally change the current location. Use this command along with `dlist`, `dshowname`, and `dmoveto`, or their synonyms `dls`, `dpwd`, and `dcd` respectively, to navigate through the directory and view its contents.

The *displayOptions* will control the output by allowing you to select the set of attributes to display. The LDAP-specific options will control how internal operations access the directory, such as dereferencing aliases or not when accessing the target entry.

SYNONYMS

`dshow`

SEE ALSO

`dlist, dmoveto`

dshown

Synonym for the `dshowname` command.

dshowname

Display the full DN of the current location or of the specified target object.

SYNTAX

`dshowname [`*object*`] [-ufn] [-ldap]`

OPTIONS

object	Specify the object whose DN you wish to display. The default is the current location.
`-ufn` `-ldap`	Display the name in the “user-friendly” format defined by RFC 1781 or in full DN format specified by LDAP. The user-friendly format removes the attribute names from the DN to leave the values separated by commas. The default is `-ldap`.

DESCRIPTION

The `dshowname` command will display the full distinguished name of the target entry. Use this command on a sequence number to see the DN of the corresponding entry in the sequence. The “user-friendly” display of the `-ufn` option is equivalent to the format used in the `ilash` command prompt.

EXAMPLE

```
[People,Siroe,com]% dshowname
ou=People, dc=Siroe,dc=com

[People,Siroe,com]% dshowname J.4 -ufn
bjensen, People, Siroe,com
```

SYNONYMS

`dshown, dpwd`

dstatus

Give information about the connection to the current directory server.

SYNTAX

```
dstatus
   [-user] [-[no]stats] [-[no]session]
   [-[no]controls] [-[no]extensions] [-[no]config]
   [-[no]syntax]
```

OPTIONS

`-user`	Display the bind DN used for the current connection.
`-session`	Display the status of the `ilash` interpreter. This includes the directory URL, the bind DN, the authentication level, the current location and the current sequence name. This is the default output when no option is specified.
`-stats`	Display the directory usage statistics if supported by the server.
`-controls`	List the OIDs of controls supported by the server. The list includes brief description of each control, if known.
`-extensions`	List the OIDs of extensions supported by the server. The list includes a brief description of each extension, if known.
`-config`	Display the configuration information published by the server. This includes the schema entry DN and information about the naming contexts on the server. Information that is not supported is left blank.
`-syntax`	Displays the list of OIDs for all attribute syntaxes used in the schema of the current directory.

DESCRIPTION

The `dstatus` command displays information about the connection and the configuration of the currently bound directory server. When used without any options, it shows the status of the `ilash` interpreter, such as the current location and the current sequence. You may specify any number of options to list all of the configuration information at one time.

EXAMPLE

```
[Siroe,com]% dstatus
Connected to         : ldap://ldap.Siroe.com:389/
Current position     : dc=Siroe,dc=com
User name            : cn=directory manager
Current sequence     : mysearch
Authentication level : simple
```

SYNONYMS

```
dinfo
```

dunbind

Close the connection to the current directory and optionally exit the `ilash` tool.

SYNTAX

```
dunbind
   [-[no]quit]
```

OPTIONS

`-quit` `-noquit`	Specify whether or not to exit the `ilash` tool after unbinding. The default is `-noquit`.

DESCRIPTION

The `dunbind` command closes the current connection. If you do not use the `-quit` option, the `ilash` tool will be in the unbound state. You will need to call the `dbind` command to bind to another server or to rebind to the same server as a different user.

SEE ALSO

`dquit, quit, dbind, drebind`

help

Synonym for the `dhelp` command.

quit

Synonym for the `dunbind -quit` command.

quit

Index